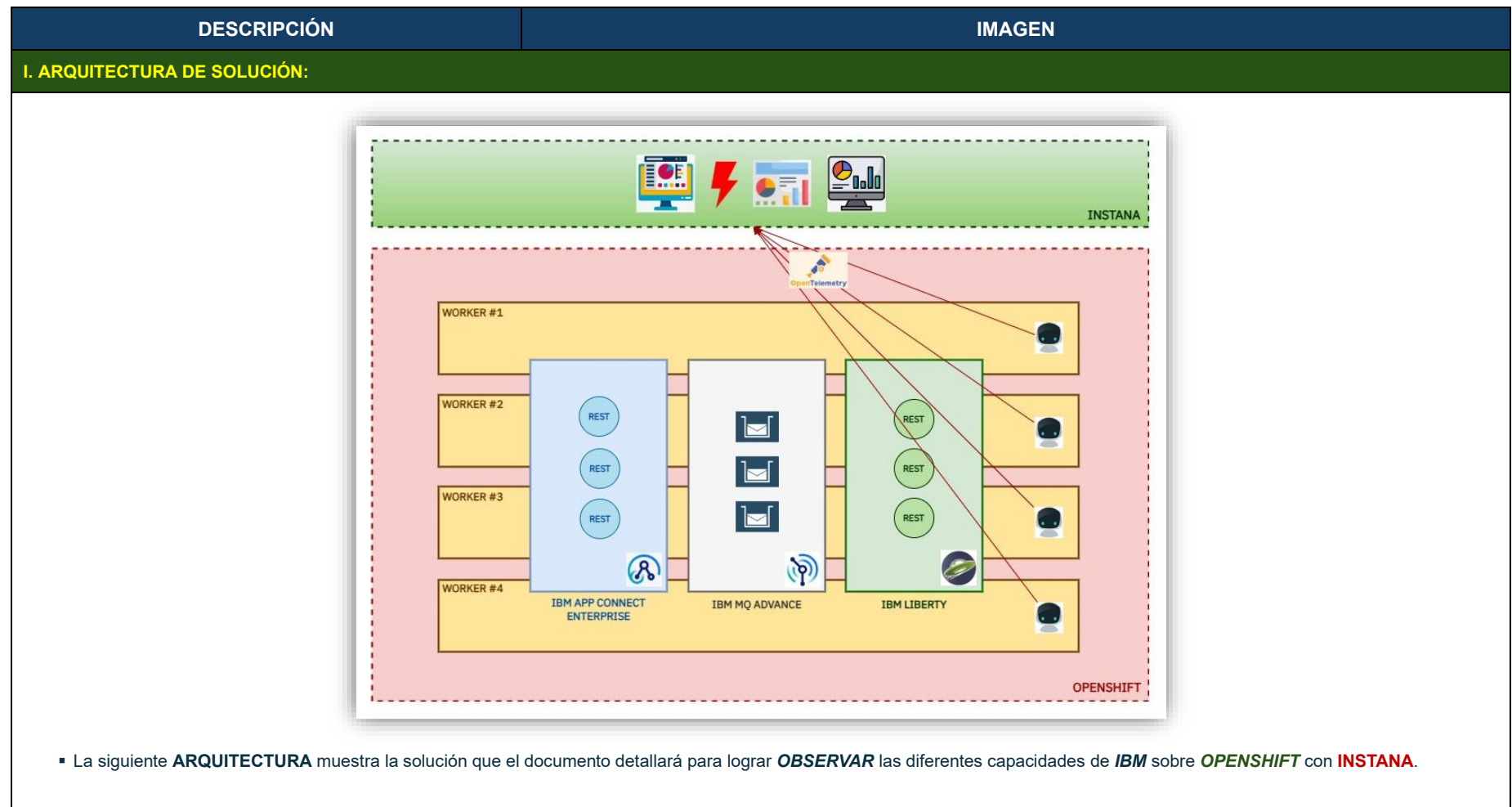




## IBM's CAPABILITY OBSERVABILITY WITH: 'INSTANA'

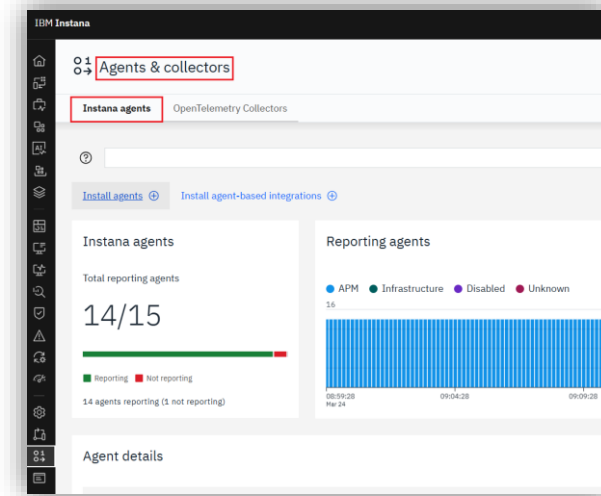
El siguiente documento detallará el **PASO** a **PASO** el manejo, configuración & recomendaciones de **OBSERVABILIDAD**, asociada a diferentes **CAPACIDADES** de **IBM** desplegadas sobre **OPENSIFT**:

- ✓ **INSTANA & OPENSIFT.**
- ✓ **IBM APP CONNECT ENTERPRISE.**
- ✓ **IBM MQ ADVANCED.**
- ✓ **IBM LIBERTY.**
- ✓ **ALERT MANAGER.**

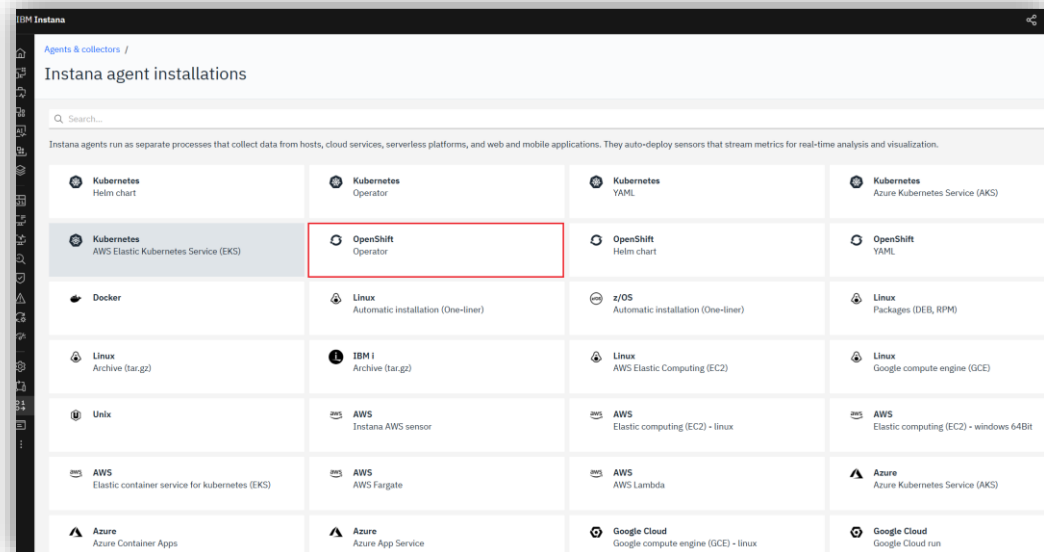


## II. INSTANA & OPENSIFT (INTEGRACIÓN):

- Antes que nada, se requiere **INTEGRAR: INSTANA & OPENSIFT**.
- Para ello se debe ingresar a la **CONSOLA** de **INSTANA** & en el **MENÚ PRINCIPAL** seleccionar: **AGENTS & COLLECTORS**.
- Luego, ir a la **PESTAÑA: INSTANA AGENTS**.



- Luego, se mostrará un montón de **AGENTES** asociados a los diferentes escenarios para la **INSTALACIÓN** de **INSTANA**
- En este caso como se va a trabajar con **OPENSIFT**, seleccionar: **OPENSIFT (OPERATOR)**.





- Luego, se procede a instalar el **CATALOG-SOURCE** asociado a **INSTANA**:

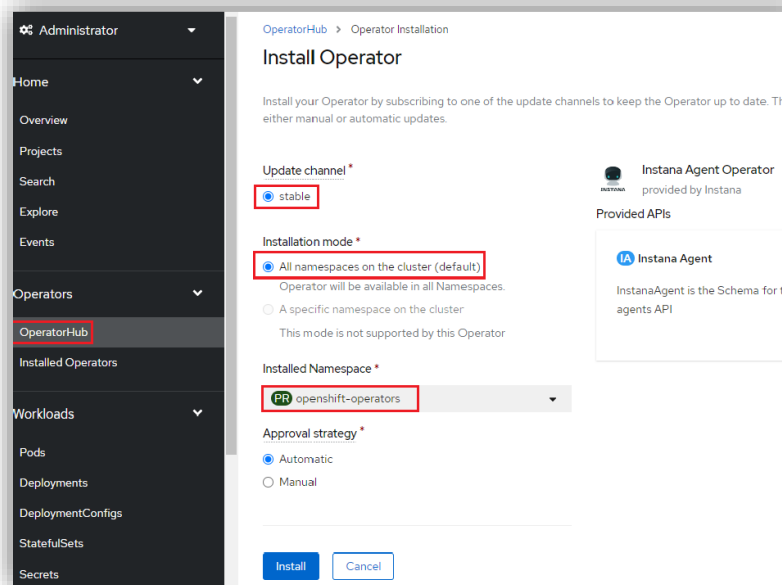
```
apiVersion: operators.coreos.com/v1alpha1
kind: CatalogSource
metadata:
  name: certified-operators
  namespace: openshift-marketplace
spec:
  displayName: Certified Operators
  image: 'registry.redhat.io/redhat/certified-operator-index:v4.18'
  publisher: Red Hat
  sourceType: grpc
  updateStrategy:
    registryPoll:
      interval: 10m
```

**IMPORTANTE:** "Esto hará que el **OPERADOR** se muestre en **OPENSIFT**, ya que por default viene **DESACTIVADO**".

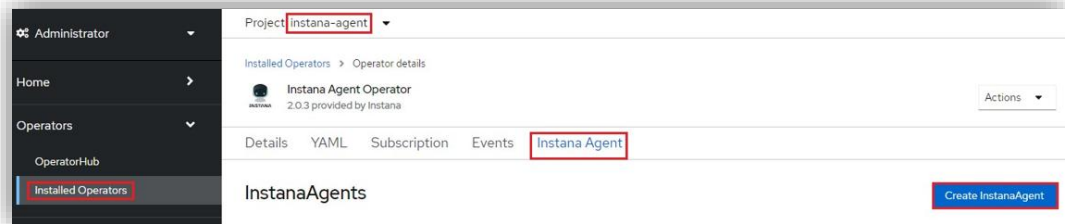
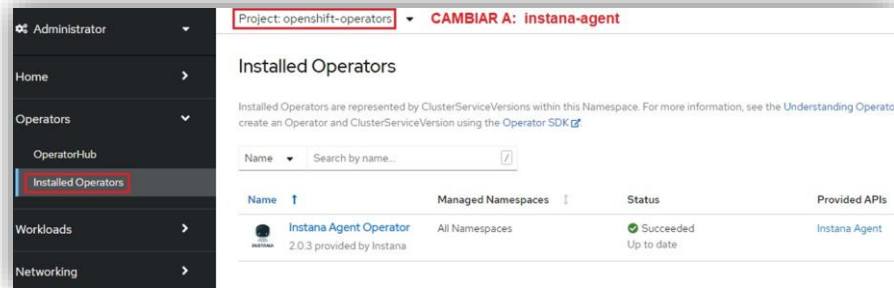
```
1 apiVersion: operators.coreos.com/v1alpha1
2 kind: CatalogSource
3 metadata:
4   name: certified-operators
5   namespace: openshift-marketplace
6 spec:
7   displayName: Certified Operators
8   image: 'registry.redhat.io/redhat/certified-operator-index:v4.18'
9   publisher: Red Hat
10  sourceType: grpc
11  updateStrategy:
12    registryPoll:
13      interval: 10m
```

- Luego, se ingresa en **OPERATOR-HUB** seleccionar el **OPERADOR** de **INSTANA** & configurarlo:

- ✓ **UPDATE CHANNEL:** *stable*
- ✓ **INSTALLATION MODE:** *All namespaces ...*
- ✓ **INSTALLED NAMESPACE:** *openshift-operators*



- Luego, ingresar en **INSTALLED-OPERATORS** & seleccionar al **NAMESPACE: instana-agent**.
- La **INSTANCIA** de **INSTANA** debe figurar en estado: **EXITOSO**.
- Luego, ingresar en la **INSTANCIA** & seleccionar la **PESTAÑA: INSTANA AGENT** & dar en **CREAR**.



- Luego, se procede a crear la **INSTANCIA** desde la plataforma de **INSTANA**.
- El **SCRIPT** utilizado será este, al cual se le ha agregado el soporte de **OPEN-TELEMETRY** & para el **DEBUG**:

```

apiVersion: instana.io/v1
kind: InstanaAgent
metadata:
  name: instana-agent
  namespace: instana-agent
spec:
  agent:
    key: ORIjrirMTj6PiA67APP16g
    downloadKey: ORIjrirMTj6PiA67APP16g
    endpointHost: ingress-coral-saas.instana.io
    endpointPort: '443'
    env:
      INSTANA_LOG_LEVEL: DEBUG
  cluster:
    name: CP4I-CRGA-CLUSTER
  opentelemetry:
    grpc:
      enabled: true
    http:
      enabled: true
  zone:
    name: CP4I-CRGA-CLUSTER

```

```

Proyecto: instana-agent
Operadores instalados > instana-agent-operatorcv2.2.8 > Detalles de InstanaAgent
IA instana-agent
Detalles YAML Recursos Eventos
1 apiVersion: instana.io/v1
2 kind: InstanaAgent
3 metadata:
4   creationTimestamp: '2026-03-24T03:30:59Z'
5   finalizers:
6     - v3.agent.instana.io/finalizer
7   generation: 1
8   managedFields: ...
102  name: instana-agent
103  namespace: instana-agent
104  resourceVersion: '93100129'
105  uid: 33b29af1-2648-4491-a293-d227a0a73c23
106 spec:
107   agent:
108     configuration_yaml: |
109       # You can leave this empty or use this to configure your instana agent.
110       # see https://docs.instana.io/setup_and_manage/host_agent/on/kubernetes/
111       downloadKey: ORIjrirMTj6PiA67APP16g
112       endpointHost: ingress-coral-saas.instana.io
113       endpointPort: '443'
114       env:
115         INSTANA_LOG_LEVEL: DEBUG
116       key: ORIjrirMTj6PiA67APP16g
117   cluster:
118     name: CP4I-CRGA-CLUSTER
119   opentelemetry:
120     grpc:
121       enabled: true
122     http:
123       enabled: true
124   zone:

```

- Luego, se procede la **CONFIGURACIÓN** del **PLUGIN** de **INSTANA** para **OPEN-TELEMETRY** a nivel de la **INSTANCIA**, agregando el siguiente bloque:

```
spec:
  agent:
    configuration_yaml: |
      com.instana.plugin.opentelemetry:
        enabled: true
        grpc:
          enabled: true
          port: 4317
        http:
          enabled: true
          port: 4318
```

**IMPORTANTE:** “Todas estas **CONFIGURACIONES** se deben realizar dentro del **NAMESPACE: instana-agent**”.

“Asegurarse que los **PODs** asociados a **INSTANA-AGENTS**, sean reiniciados ante la modificación de la **INSTANCIA** de **INSTANA**, sino hacerlo **MANUALMENTE**”.

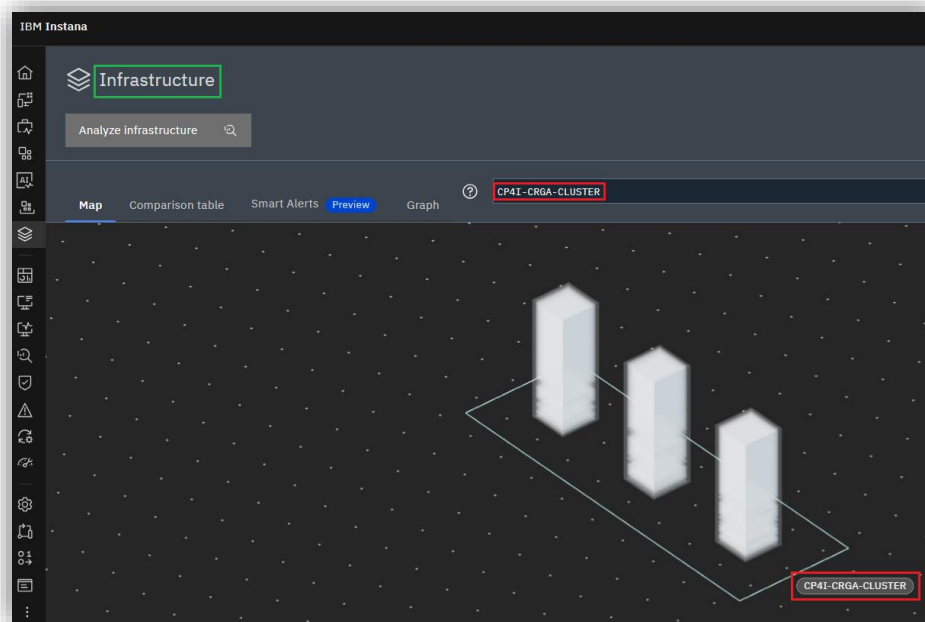
```
generation: 6
managedFields: ...
name: instana-agent
namespace: instana-agent
resourceVersion: '96656864'
uid: 33b29af1-2648-4491-a293-d227a0a73c23
spec:
  agent:
    configuration_yaml: |
      # You can leave this empty or use this to configure your instana agent.
      # See https://docs.instana.io/setup_and_manage/host_agent/on/kubernetes/
      com.instana.plugin.opentelemetry:
        enabled: true
        grpc:
          enabled: true
          port: 4317
        http:
          enabled: true
          port: 4318
    downloadKey: ORI3rirMTj6P1A67APP16g
    endpointHost: ingress-coral-saas.instana.io
    endpointPort: '443'
```

- Luego, se valida que tanto los **AGENTES** como los **SENSORES** de **INSTANA** estén en estado: **RUNNING**.

**IMPORTANTE:** “El número de **AGENTES & SENSORES** es equivalente al **N#** de **WORKER-NODES** del **CLÚSTER**”.

Nombre	Estado	Listo	Se reinicia	Propietario	Memoria	CPU
instana-agent-2f522	Running	1/1	0	DS instana-agent	546,8 MIB	0,299 núcleos
instana-agent-9xdzg	Running	1/1	0	DS instana-agent	615,6 MIB	0,334 núcleos
instana-agent-zgzzs	Running	1/1	0	DS instana-agent	549,0 MIB	0,317 núcleos
instana-agent-k8sensor-5dd4bf4f7-ptxd9	Running	1/1	0	RS instana-agent-k8sensor-5dd4bf4f7	597,1 MIB	0,087 núcleos
instana-agent-k8sensor-5dd4bf4f7-abbar	Running	1/1	1	RS instana-agent-k8sensor-5dd4bf4f7	572,8 MIB	0,107 núcleos
instana-agent-k8sensor-5dd4bf4f7-ertkz	Running	1/1	1	RS instana-agent-k8sensor-5dd4bf4f7	603,7 MIB	0,075 núcleos

- Luego, desde el **MENÚ GENERAL** se ingresa en: **INFRAESTRUCTURA**.
- Ahí se filtra ingresando el nombre del **CLÚSTER**:  
 ✓ **CP4I-CRGA-CLUSTER**



- Finalmente, en la **PESTAÑA: COMPARISON TABLE** & desde aquí se podrá ver:

- ✓ **CANTIDAD DE WORKER NODES.**
- ✓ **NOMBRES & IPs DE WORKER NODES**
- ✓ **RECURSOS DE WORKER NODES**
- ✓ **ESTADO DE WORKER-NODES.**

Zone	Name	IP addresses	OS	Type	#CPUs	CPU usage	Memory	Memory used	Health
-	ibm-gcpa-worker-0-02-gcpa-ibm01-0-0014-techno.ibm.com	192.168.21.2, 10.129.2.2, 10.10.10.21	5.14.0-427.105.1.a99_4.a86_44 (amd64)	-	16	33%	62,76 GiB	53%	OK
-	ibm-gcpa-worker-0-02-gcpa-ibm01-0-0014-techno.ibm.com	192.168.22.2, 10.131.0.2, 10.10.10.22	5.14.0-427.105.1.a99_4.a86_44 (amd64)	-	16	39%	62,76 GiB	54%	OK
-	ibm-gcpa-worker-0-02-gcpa-ibm01-0-0014-techno.ibm.com	192.168.23.2, 10.128.2.2, 10.10.10.23	5.14.0-427.105.1.a99_4.a86_44 (amd64)	-	16	32%	62,76 GiB	56%	OK

### III. IBM APP CONNECT ENTERPRISE: (CONFIGURACIÓN)

- Se procede dentro de la **INSTANCIA** del **INTEGRATION-RUNTIME** de **ACE**, a ingresar el bloque del **SOPORTE** para: **OPEN-TELEMETRY**:

```
spec:
  telemetry:
    tracing:
      openTelemetry:
        enabled: true
```

```
122  yaml:
123  - 'https://inst-ace-dashboard-dash.integration:3443/v1/directories/Dummy
124  configurations:
125  - server.conf.yaml
126  - dummy-gestion-empleados-ir-adminssl
127  license:
128  accept: true
129  license: L-KPRV-AUG9NC
130  use: CloudPakForIntegrationProduction
131  logFormat: basic
132  replicas: 1
133  telemetry:
134  tracing:
135  openTelemetry:
136  enabled: true
137  template:
138  spec:
139  containers:
140  - name: runtime
141  resources:
142  requests:
143  cpu: 300m
144  memory: 368Mi
```

- Luego, considerando que el **SERVICIO** implementado en **ACE** desplegado se llama: **dummy-gestion-empleados** en el **NAMESPACE**: **integration**.

- Se procede a ejecutar el **COMANDO** para obtener el valor exacto:

```
$ oc get services -n integration | grep dummy-gestion-empleados*
```

- Aquí se aprecia que el nombre del **RECURSO SERVICE** trabajado es: **dummy-gestion-empleados-ir**, con esto se procede a definir el **PATRÓN** a utilizar:

```
ResourceManagers:
  ExceptionLog:
    enabled: true
  OpenTelemetryManager:
    exporterOtlpGrpcEndpoint: instana-agent.instana-agent.svc:4317
    openTelemetryEnabled: true
    openTelemetryHostName: <RECURSO-SERVICE-ACE>.svc.cluster.local
    openTelemetryScope: extended
    openTelemetryServiceName: <RECURSO-SERVICE-ACE>
```

```
AzureAD+CesarRicardoGuerraAr@IBM-PF5R3G8K MINGW64 ~
$ oc get services -n integration | grep dummy-gestion-empleados*
dummy-gestion-empleados-ir          ClusterIP 172.30.118.192 <none> 7800/TCP,7843/TCP,7600/TCP
dummy-gestion-empleados-ir-metrics ClusterIP 172.30.136.83 <none> 9483/TCP
```

Configuration

Name	Type
dummy-gestion-empleados-ir-adminssl	REST Admin SSL files
dummy-aggregation-rest-start-mode	WorkdirOverride
server.conf.yaml	server.conf.yaml
traceslogging	server.conf.yaml

Update configuration

When you update a configuration that's used by deployed BAR files, those BAR files are restarted automatically.

Type: server.conf.yaml

Name: server.conf.yaml

Description (optional): CONFIGURACIÓN PARA SOPORTE DE OPENTELEMETRY

```
server.conf.yaml
1 OpenTelemetryManager:
2   openTelemetryEnabled: true
3   openTelemetryScope: 'standard'
4   openTelemetryServiceName: 'dummy-ges
5   openTelemetryHostName: 'instana-ager
```

▪ Así quedaría para su registro desde: **ACE-DASHBOARD**, con el objetivo de **OVERRIDE** el: **server.conf.yaml** del **INTEGRATION-RUNTIME**.

▪ Este sería la **CONFIGURACIÓN** final en este caso a utilizar:

ResourceManagers:

ExceptionLog:

enabled: true

OpenTelemetryManager:

exporterOtlpGrpcEndpoint: *instana-agent.instana-agent.svc:4317*

openTelemetryEnabled: true

openTelemetryHostName: *dummy-gestion-empleados-ir.integration.svc.cluster.local*

openTelemetryScope: *extended*

openTelemetryServiceName: *dummy-gestion-empleados-ir*

▪ Luego, ingresar en el **INTEGRATION-RUNTIME** en este caso: **dummy-gestion-empleados** & editar el **YAML** agregando la referencia al **CONFIGURATOR** recientemente creado, este caso: **server.conf.yaml**

**IMPORTANTE:** "Es necesario el **REINICIO** del **POD** asociado al **INTEGRATION-RUNTIME**, para que coja los cambios".

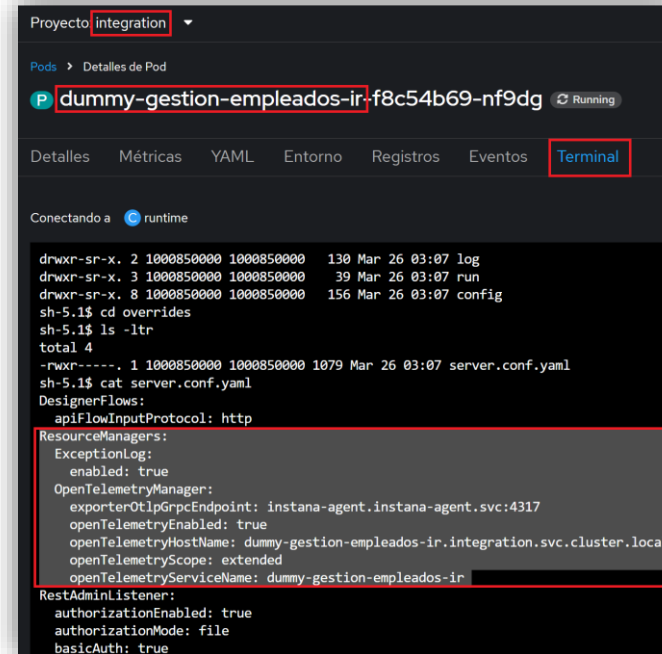
```
117 appconnect.ibm.com/designerapiflow: 'false'
118 appconnect.ibm.com/designerapiflow: 'false'
119 appconnect.ibm.com/toolkitflow: 'true'
120 backup.appconnect.ibm.com/component: integrationruntime
121 spec:
122   barURL:
123     - 'https://inst-ace-dashboard-dash.integration:3443/v1/directories/D
124   configurations:
125     - server.conf.yaml
126     - dummy-gestion-empleados-ir-adminssl
127   license:
128     accept: true
129     license: L-KPRV-AUG9NC
130     use: CloudPakForIntegrationProduction
131   logFormat: basic
132   replicas: 1
133   telemetry:
```

▪ Luego, ingresar en el **POD** del **INTEGRATION-RUNTIME** desde el **TERMINAL**, para validar la **CONFIGURACIÓN** realizada.

▪ Aquí ingresamos los **COMANDOS** dentro del **POD**:

```
$ ls -ltr  
$ cd ace-server  
$ ls -ltr  
$ cd overrides  
$ cat server.conf.yaml
```

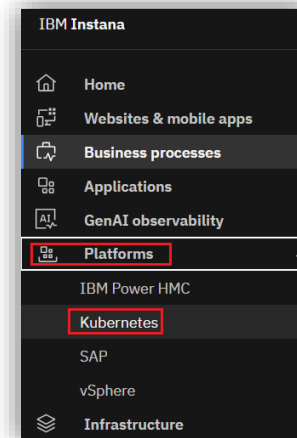
**IMPORTANTE:** “Este **ARCHIVO** deberá mostrar la **CONFIGURACIÓN** registrada previamente para el **OPEN-TELEMETRY** dentro del **INTEGRATION-RUNTIME**, tal como se muestra en la **IMAGEN**”.



The screenshot shows a terminal window with the following content:

```
Proyecto integration  
Pods > Detalles de Pod  
P dummy-gestion-empleados-ir-f8c54b69-nf9dg Running  
Detalles Métricas YAML Entorno Registros Eventos Terminal  
Conectando a runtime  
drwxr-sr-x. 2 1000850000 1000850000 130 Mar 26 03:07 log  
drwxr-sr-x. 3 1000850000 1000850000 39 Mar 26 03:07 run  
drwxr-sr-x. 8 1000850000 1000850000 156 Mar 26 03:07 config  
sh-5.1$ cd overrides  
sh-5.1$ ls -ltr  
total 4  
-rwxr-----. 1 1000850000 1000850000 1079 Mar 26 03:07 server.conf.yaml  
sh-5.1$ cat server.conf.yaml  
DesignerFlows:  
apiFlowInputProtocol: http  
ResourceManagers:  
ExceptionLog:  
  enabled: true  
OpenTelemetryManager:  
  exporterOtlpGrpcEndpoint: instana-agent.instana-agent.svc:4317  
  openTelemetryEnabled: true  
  openTelemetryHostName: dummy-gestion-empleados-ir.integration.svc.cluster.local  
  openTelemetryScope: extended  
  openTelemetryServiceName: dummy-gestion-empleados-ir  
RestAdminListener:  
  authorizationEnabled: true  
  authorizationMode: file  
  basicAuth: true
```

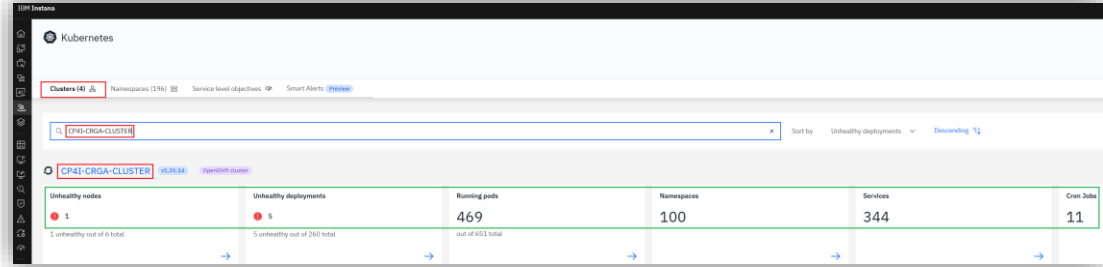
▪ Luego, desde el **MENÚ PRINCIPAL** seleccionar: **PLATFORMS** / **KUBERNETES**.



- Luego, desde la **PESTAÑA: CLÚSTERS** ingresar el nombre del **CLÚSTER** asignado al **OPENSIFT** utilizado.

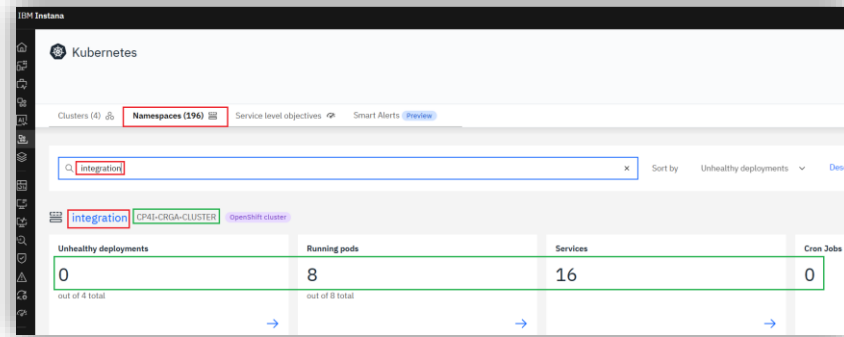
- En este caso se muestra: **CP4I-CRGA-CLUSTER**.

**IMPORTANTE:** "El **NOMBRE** configurado se encuentra en la **YAMEL** de la **INSTANCIA** de **INSTANA** creada".



- Luego, desde la **PESTAÑA: NAMESPACES** ingresar en nombre del **NAMESPACE** donde se desplegó el: **INTEGRATION-RUNTIME** del **ACE**.

- En este caso ingresar: **integration**.

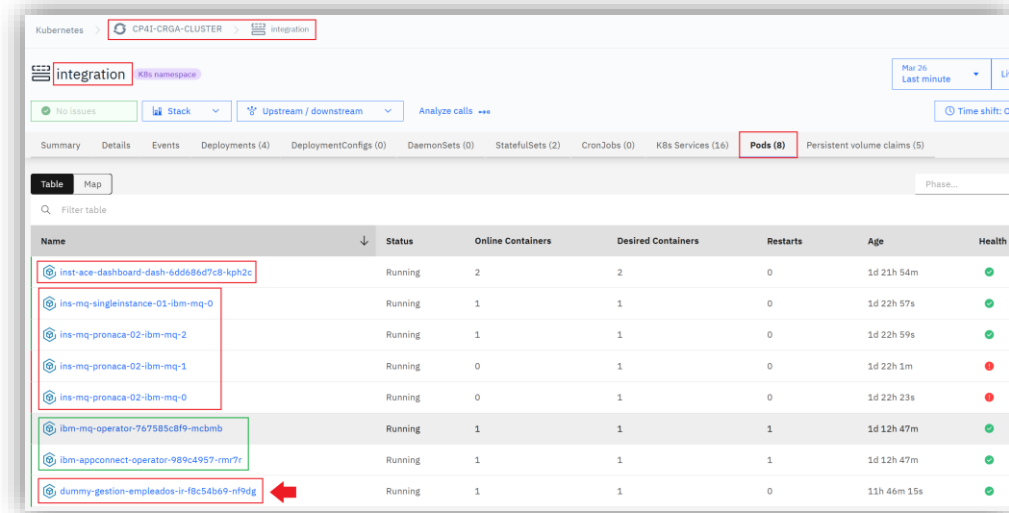


- Luego, estando dentro se mostrarán diferentes **PESTAÑAS**, asociadas a los diferentes **RECURSOS** manejados por **OPENSIFT**.

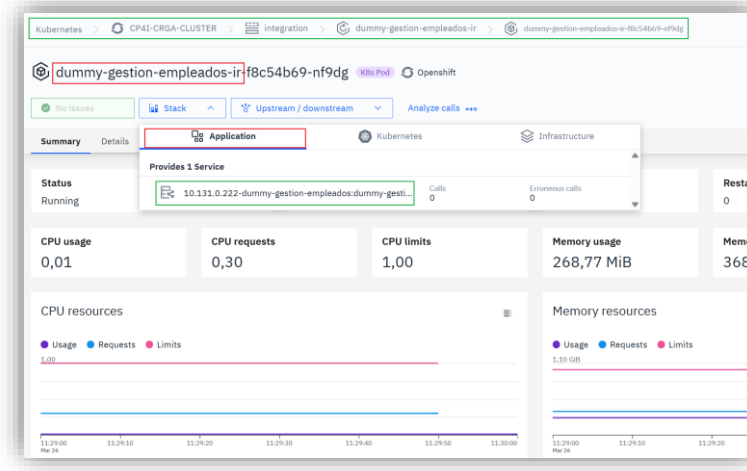
- Se ingresa en la **PESTAÑA: PODS** & se mostrarán los **PODs** asociados al **SOFTWARE** instalado en este **NAMESPACE**:

- ✓ **OPERADORES:** ACE, MQ.
- ✓ **APLICACIONES:** ACE, DASHBOARD, MQ.

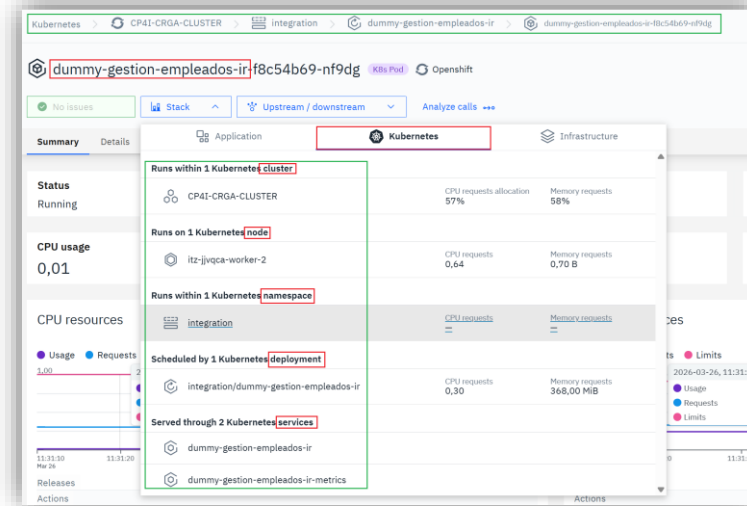
**IMPORTANTE:** "Se debe identificar el nombre en este caso del **POD** asociado a **INTEGRATION-RUNTIME** de **ACE**".



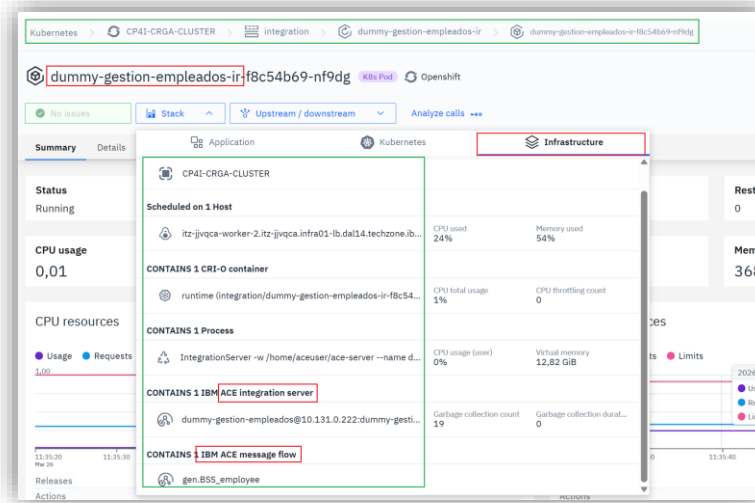
- Luego, se ingresa al **POD** asociado al **INTEGRATION-RUNTIME** en este caso: **dummy-gestion-empleados-ir-xxxxx**.
- Aquí se aprecia que los **INDICADORES** asociados a las pruebas realizadas contra el **SERVICIO** de **ACE**, se han cargado.
- Al seleccionar en **STACK** & aparecerán las opciones: **APPLICATION**, **KUBERNETES**, **INFRAESTRUCTURE**.
- Después, seleccionar: **STACK / APPLICATION** & se mostrarán el **SERVICIO** de **ACE** identificado.



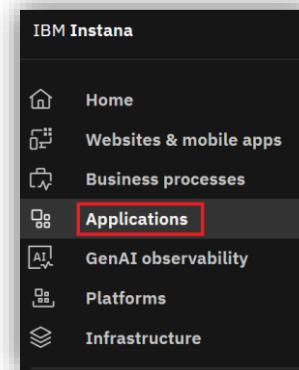
- Otra opción es seleccionar: **STACK / KUBERNETES** & se mostrarán los **RECURSOS** de **OPENSHIFT** asociados al **SERVICIO** de **ACE**.



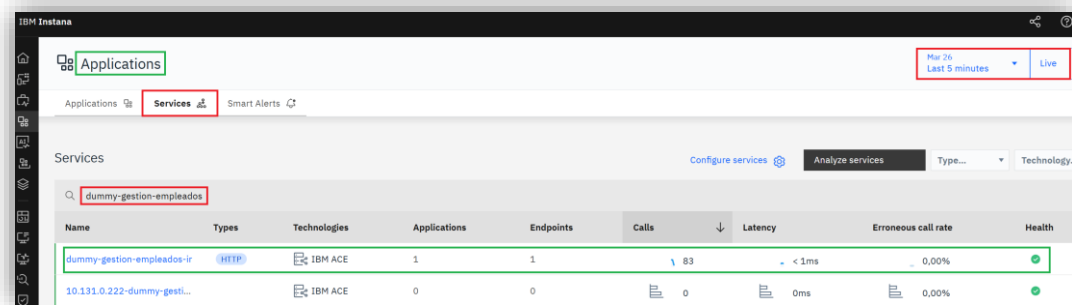
- Otra opción es seleccionar: **STACK** / **INFRASTRUCTURE** & se mostrarán los **RECURSOS** de **INFRA** asociados al **SERVICIO** de **ACE**.



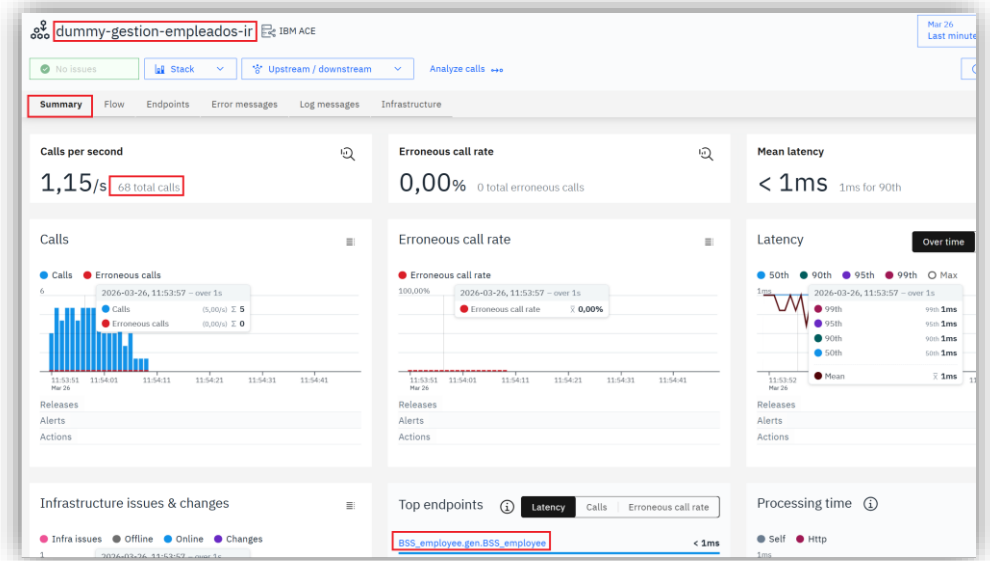
- Luego, del **MENÚ PRINCIPAL** se procede a seleccionar: **APPLICATIONS**.



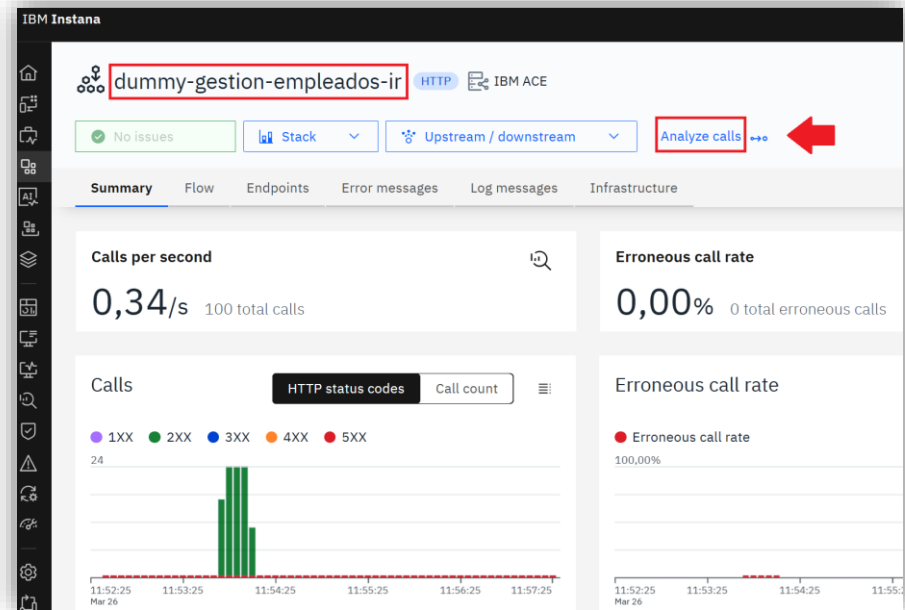
- Luego, se procede a seleccionar la **PESTAÑA: SERVICES** & ingresar en el filtro el nombre del **INTEGRATION-RUNTIME**.
- En este caso: **dummy-gestion-empleados-ir**.
- Se procederá a mostrar el resultado de la **TECNOLOGÍA** asociada al **SERVICIO** & el resultado en caliente de los **REQUEST** recibidos.



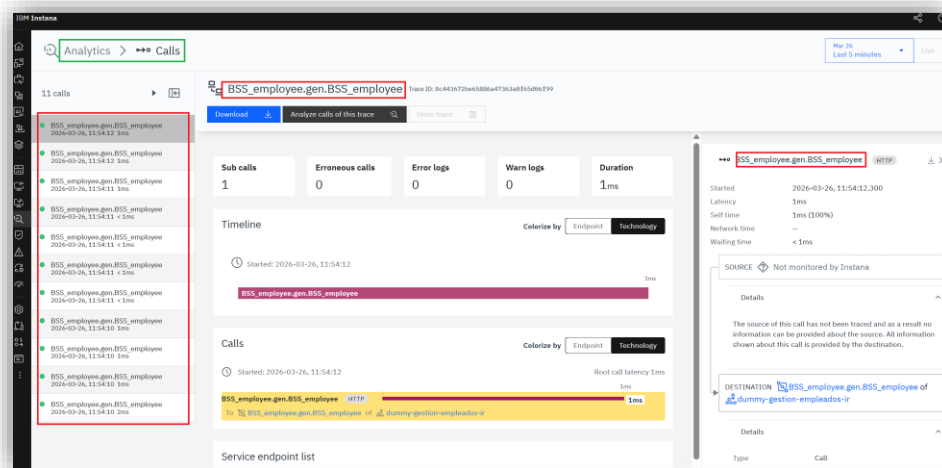
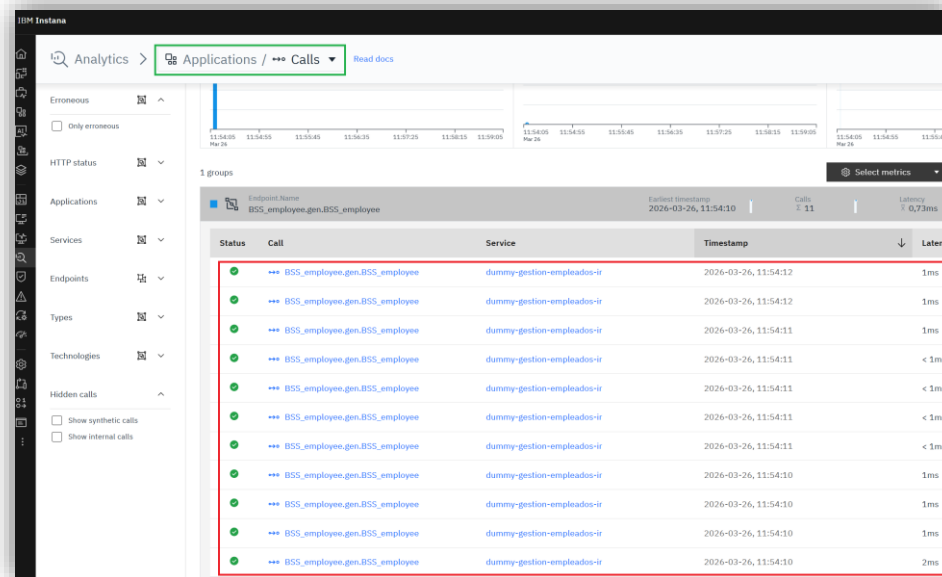
- Luego, ingresando en el **SERVICE** se muestra un **DASHBOARD** completo de **INDICADORES**.
- Estos **INDICADORES** están asociados a las **TRAZAS** de las pruebas ejecutadas del **SERVICIO** del **ACE**.



- Luego, se procede a seleccionar: **ANALYZE CALLS**, con el objetivo de obtener todo el **DETALLE** de los **REQUEST** obtenidos.
- En el **DASHBOARD** se muestra que han sido procesadas **100 REQUEST (CALLS)**, que en mi caso se enviaron por **POSTMAN**.



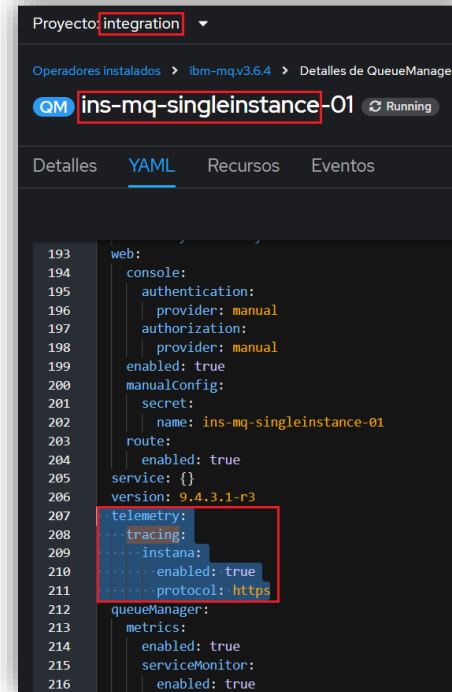
- Finalmente, se muestra el listado de **REQUEST** procesados por el **SERVICIO** del **ACE** dentro del **INTEGRATION-RUNTIME**.
- Así mismo, al seleccionar cualquiera se procede a obtener todo el **DETALLE** del mensaje.



#### IV. IBM MQ ADVANCED: (CONFIGURACIÓN)

- Se procede dentro de la **INSTANCIA** utilizada del **MQM** de **MQ**, a ingresar & agregar el bloque de soporte para: **OPEN-TELEMETRY**:

```
spec:  
  telemetry:  
    tracing:  
      instana:  
        enabled: true  
        protocol: https
```



Proyecto: integration

Operadores instalados > ibm-mq.v3.6.4 > Detalles de QueueManager

**ins-mq-singleinstance-01** Running

Detalles **YAML** Recursos Eventos

```
193 web:  
194   console:  
195     authentication:  
196       provider: manual  
197     authorization:  
198       provider: manual  
199     enabled: true  
200     manualConfig:  
201       secret:  
202         name: ins-mq-singleinstance-01  
203     route:  
204       enabled: true  
205     service: {}  
206     version: 9.4.3.1-r3  
207     telemetry:  
208       tracing:  
209         instana:  
210           enabled: true  
211           protocol: https  
212 queueManager:  
213   metrics:  
214     enabled: true  
215     serviceMonitor:  
216       enabled: true
```

- Luego, se requiere **GENERAR** un **CERTIFICADO / PASSWORD** para el **AGENTE** de **INSTANA** asociado al **MQM**.

- Debido a ello, se requiere ejecutar los siguientes **COMANDOS**:

```
$ mkdir certificado-mq
$ cd certificado-mq
$ export MSYS_NO_PATHCONV=1
$ openssl req -newkey rsa:2048 -nodes -keyout server.key -subj "/CN=mq queuemanager/OU=ibm mq" -x509 -days 3650 -out server.crt
$ ls -ltr
```

```
$ openssl req -newkey rsa:2048 -nodes -keyout application.key -subj "/CN=instana-agent/OU=app team1" -x509 -days 3650 -out application.crt
$ ls -ltr
```

```
$ openssl pkcs12 -export -out application.p12 -inkey application.key -in application.crt -passout pass:123456
$ ls -ltr
```

```
$ keytool -importkeystore -srckeystore application.p12 -srcstoretype pkcs12 -destkeystore application.jks -deststoretype JKS -srcstorepass 123456 -deststorepass 123456 -noprompt
```

```
$ keytool -changealias -alias "1" -destalias "instana" -keypass 123456 -keystore application.jks -storepass 123456 -noprompt
```

```
$ keytool -importcert -file server.crt -keystore application.jks -storepass 123456 -alias myca -noprompt
```

```
AzureAD+CesarRicardoGuerra@IBH-PF5R3G8K MINGW64 ~
$ mkdir certificado-mq

AzureAD+CesarRicardoGuerra@IBH-PF5R3G8K MINGW64 ~
$ cd certificado-mq/

AzureAD+CesarRicardoGuerra@IBH-PF5R3G8K MINGW64 ~/certificado-mq
$ export MSYS_NO_PATHCONV=1

AzureAD+CesarRicardoGuerra@IBH-PF5R3G8K MINGW64 ~/certificado-mq
$ openssl req -newkey rsa:2048 -nodes -keyout server.key -subj "/CN=mq queuemanager/OU=ibm mq" -x509 -days 3650 -out server.crt
.....
+++++
-----

AzureAD+CesarRicardoGuerra@IBH-PF5R3G8K MINGW64 ~/certificado-mq
$ ls -ltr
total 8
-rw-r--r-- 1 AzureAD+CesarRicardoGuerra 4096 1732 mar. 26 15:19 server.key
-rw-r--r-- 1 AzureAD+CesarRicardoGuerra 4096 1196 mar. 26 15:19 server.crt

AzureAD+CesarRicardoGuerra@IBH-PF5R3G8K MINGW64 ~/certificado-mq
$ openssl req -newkey rsa:2048 -nodes -keyout application.key -subj "/CN=instana-agent/OU=app team1" -x509 -days 3650 -out application.crt
.....
+++++
-----

AzureAD+CesarRicardoGuerra@IBH-PF5R3G8K MINGW64 ~/certificado-mq
$ ls -ltr
total 16
-rw-r--r-- 1 AzureAD+CesarRicardoGuerra 4096 1732 mar. 26 15:19 server.key
-rw-r--r-- 1 AzureAD+CesarRicardoGuerra 4096 1196 mar. 26 15:19 server.crt
-rw-r--r-- 1 AzureAD+CesarRicardoGuerra 4096 1732 mar. 26 15:23 application.key
-rw-r--r-- 1 AzureAD+CesarRicardoGuerra 4096 1200 mar. 26 15:23 application.crt

AzureAD+CesarRicardoGuerra@IBH-PF5R3G8K MINGW64 ~/certificado-mq
$ openssl pkcs12 -export -out application.p12 -inkey application.key -in application.crt -passout pass:123456

AzureAD+CesarRicardoGuerra@IBH-PF5R3G8K MINGW64 ~/certificado-mq
$ ls -ltr
total 20
-rw-r--r-- 1 AzureAD+CesarRicardoGuerra 4096 1732 mar. 26 15:19 server.key
-rw-r--r-- 1 AzureAD+CesarRicardoGuerra 4096 1196 mar. 26 15:19 server.crt
-rw-r--r-- 1 AzureAD+CesarRicardoGuerra 4096 1732 mar. 26 15:23 application.key
-rw-r--r-- 1 AzureAD+CesarRicardoGuerra 4096 1200 mar. 26 15:23 application.crt
-rw-r--r-- 1 AzureAD+CesarRicardoGuerra 4096 2595 mar. 26 15:34 application.p12
```

```
AzureAD+CesarRicardoGuerra@IBH-PF5R3G8K MINGW64 ~/certificado-mq
$ keytool -importkeystore -srckeystore application.p12 -srcstoretype pkcs12 -destkeystore application.jks -deststoretype JKS -srcstorepass 123456 -deststorepass 123456 -noprompt
Importing keystore application.p12 to application.jks...
Entry for alias 1 successfully imported.
Import command completed: 1 entries successfully imported, 0 entries failed or cancelled

Warning:
The JKS keystore uses a proprietary format. It is recommended to migrate to PKCS12 which is an industry standard format using "keytool -importkeystore -srckeystore application.jks -destkeystore application.jks -storepass 12345678 -noprompt"

AzureAD+CesarRicardoGuerra@IBH-PF5R3G8K MINGW64 ~/certificado-mq
$ keytool -changealias -alias "1" -destalias "instana" -keypass 12345678 -keystore application.jks -storepass 12345678 -noprompt
keytool error: java.io.IOException: Keystore was tampered with, or password was incorrect

AzureAD+CesarRicardoGuerra@IBH-PF5R3G8K MINGW64 ~/certificado-mq
$ keytool -changealias -alias "1" -destalias "instana" -keypass 123456 -keystore application.jks -storepass 123456 -noprompt

Warning:
The JKS keystore uses a proprietary format. It is recommended to migrate to PKCS12 which is an industry standard format using "keytool -importkeystore -srckeystore application.jks -destkeystore application.jks -storepass 12345678 -noprompt"

AzureAD+CesarRicardoGuerra@IBH-PF5R3G8K MINGW64 ~/certificado-mq
$ keytool -importcert -file server.crt -keystore application.jks -storepass 123456 -alias myca -noprompt
Certificate was added to keystore

Warning:
The JKS keystore uses a proprietary format. It is recommended to migrate to PKCS12 which is an industry standard format using "keytool -importkeystore -srckeystore application.jks -destkeystore application.jks -storepass 12345678 -noprompt"

AzureAD+CesarRicardoGuerra@IBH-PF5R3G8K MINGW64 ~/certificado-mq
$ ls -ltr
total 24
-rw-r--r-- 1 AzureAD+CesarRicardoGuerra 4096 1732 mar. 26 15:19 server.key
-rw-r--r-- 1 AzureAD+CesarRicardoGuerra 4096 1196 mar. 26 15:19 server.crt
-rw-r--r-- 1 AzureAD+CesarRicardoGuerra 4096 1732 mar. 26 15:23 application.key
-rw-r--r-- 1 AzureAD+CesarRicardoGuerra 4096 1200 mar. 26 15:23 application.crt
-rw-r--r-- 1 AzureAD+CesarRicardoGuerra 4096 2595 mar. 26 15:34 application.p12
-rw-r--r-- 1 AzureAD+CesarRicardoGuerra 4096 3038 mar. 26 20:23 application.jks
```

- Luego, desde la misma **UBICACIÓN:** (*directorio: certificado-mq*) & con acceso a ejecutar **OPENSIFT CLIENT (OC)**, se procede a crear un **SECRET** en **OPENSIFT**, con las credenciales de acceso previamente creadas.

- Ejecutar el siguiente **COMANDO:**

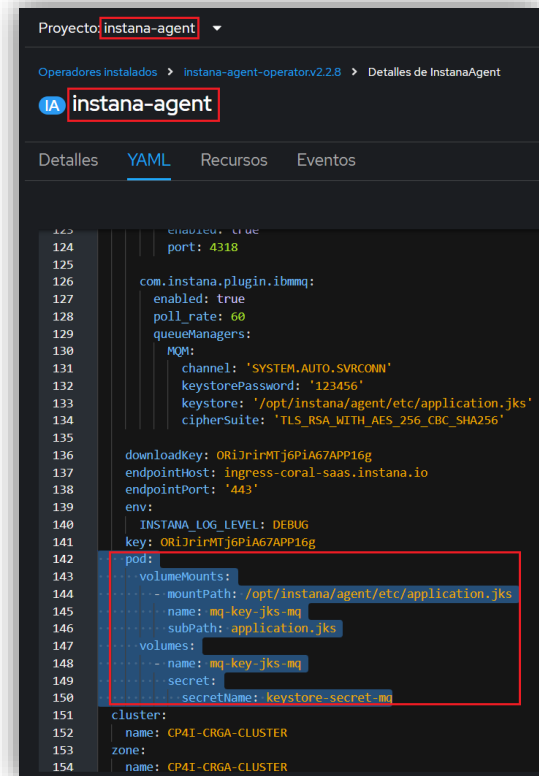
```
$ oc create secret generic keystore-secret-mq --from-file=./application.jks -n instana-agent  
=> secret/keystore-secret-mq created
```

```
AzureAD+CesarRicardoGuerraAr@IBM-PF5RJG8K MINGW64 ~/certificado-mq  
$ oc create secret generic keystore-secret-mq --from-file=./application.jks -n instana-agent  
secret/keystore-secret-mq created
```

- Luego, es necesario ingresar en el **NAMESPACE: instana-agent** & en la **INSTANCIA** creada, agregar este **BLOQUE**.

- Este **BLOQUE** se conecta con el **SECRET** previamente ejecutado, asociado internamente a los **CERTIFICADOS**.

```
agent:  
  pod:  
    volumeMounts:  
      - mountPath: /opt/instana/agent/etc/application.jks  
        name: mq-key-jks-mq  
        subPath: application.jks  
    volumes:  
      - name: mq-key-jks-mq  
        secret:  
          secretName: keystore-secret-mq
```



```
Proyecto: instana-agent  
Operadores Instalados > instana-agent-operatorv2.2.8 > Detalles de InstanaAgent  
IA instana-agent  
Detalles YAML Recursos Eventos  
122 enabled: true  
123 port: 4318  
124  
125  
126 com.instana.plugin.ibmmq:  
127   enabled: true  
128   poll_rate: 60  
129   queueManagers:  
130     MQM:  
131       channel: 'SYSTEM.AUTO.SVRCONN'  
132       keystorePassword: '123456'  
133       keystore: '/opt/instana/agent/etc/application.jks'  
134       cipherSuite: 'TLS_RSA_WITH_AES_256_CBC_SHA256'  
135  
136 downloadKey: ORiJrirMTj6P1A67APP16g  
137 endpointHost: ingress-coral-saas.instana.io  
138 endpointPort: '443'  
139 env:  
140   INSTANA_LOG_LEVEL: DEBUG  
141   key: ORiJrirMTj6P1A67APP16g  
142   pod:  
143     volumeMounts:  
144       - mountPath: /opt/instana/agent/etc/application.jks  
145         name: mq-key-jks-mq  
146         subPath: application.jks  
147     volumes:  
148       - name: mq-key-jks-mq  
149         secret:  
150           secretName: keystore-secret-mq  
151 cluster:  
152   name: CP4I-CRGA-CLUSTER  
153   zone:  
154   name: CP4I-CRGA-CLUSTER
```

- Luego, dentro de la misma **INSTANCIA** de **INSTANA**: **instana-agent**, agregar este **BLOQUE** en el: **configuration\_yaml**:

```
com.instana.plugin.ibmmq:
  enabled: true
  poll_rate: 60
  queueManagers:
    MQM:
      channel: 'SYSTEM.AUTO.SVRCONN'
      keystorePassword: '123456'
      keystore: '/opt/instana/agent/etc/application.jks'
      cipherSuite:
        'TLS_RSA_WITH_AES_256_CBC_SHA256'
```

```
109 resourceVersion: '98153360'
110 uid: 39b29af1-2648-4491-a293-d227a0a73c23
111 spec:
112   agent:
113     configuration_yaml: |
114       # You can leave this empty or use this to configure your instana agent.
115       # See https://docs.instana.io/setup_and_manage/host_agent/on/kubernetes/
116
117     com.instana.plugin.opentelemetry:
118       enabled: true
119       grpc:
120         enabled: true
121         port: 4317
122       http:
123         enabled: true
124         port: 4318
125
126     com.instana.plugin.ibmmq:
127       enabled: true
128       poll_rate: 60
129       queueManagers:
130         MQM:
131           channel: 'SYSTEM.AUTO.SVRCONN'
132           keystorePassword: '123456'
133           keystore: '/opt/instana/agent/etc/application.jks'
134           cipherSuite: 'TLS_RSA_WITH_AES_256_CBC_SHA256'
```

- Luego, es necesario **REINICIAR** los **PODs** asociados a los **AGENTES** de **INSTANA**:

```
$ oc rollout restart daemonset instana-agent -n instana-agent
```

```
AzureAD+CesarRicardoGuerraAr@IBM-PF5RJG8K MINGW64 ~/certificado-mq
$ oc rollout restart daemonset instana-agent -n instana-agent
daemonset.apps/instana-agent restarted
```

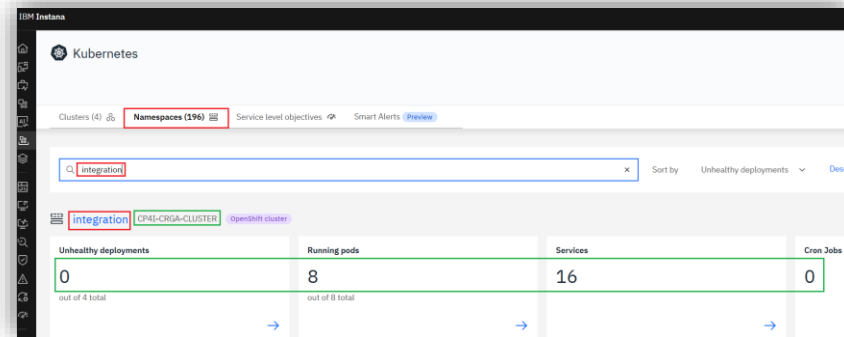
- Luego, desde la **PESTAÑA**: **CLÚSTERS** ingresar el nombre del **CLÚSTER** asignado al **OPENSIFT** utilizado.

- En este caso se muestra: **CP4I-CRGA-CLUSTER**.

**IMPORTANTE:** "Este se encuentra dentro del **YAMEL** de la **INSTANCIA** de **INSTANA** creada".

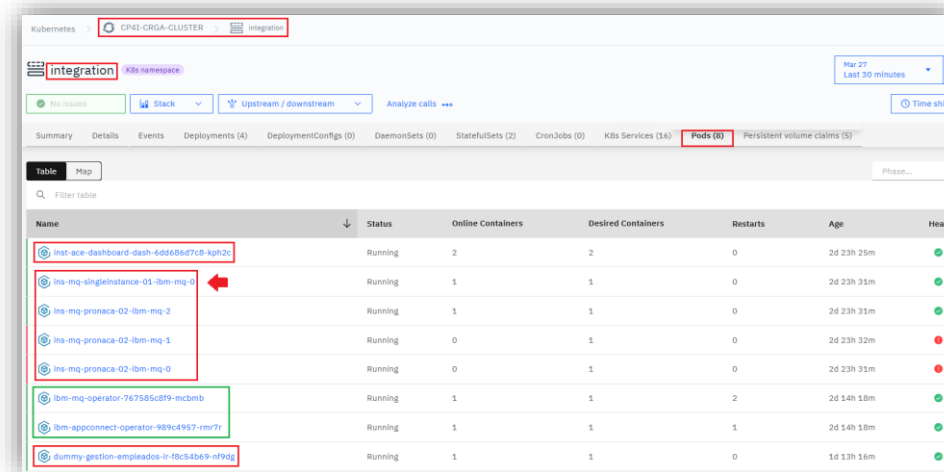
Unhealthy nodes	Unhealthy deployments	Running pods	Namespaces	Services	Cron Jobs
1	5	469	100	344	11
1 unhealthy out of 6 total	5 unhealthy out of 260 total	out of 603 total			

- Luego, desde la **PESTAÑA: NAMESPACES** ingresar en nombre del **NAMESPACE** donde se desplegó el: **MQM** del **MQ**.
- En este caso ingresar: **integration**.

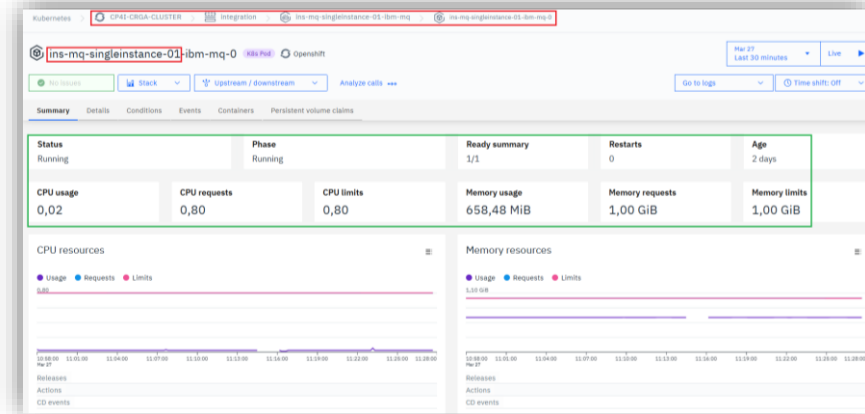


- Luego, se mostrarán las diferentes **PESTAÑAS**, asociadas a los diferentes **RECURSOS** manejados por **OPENSIFT**.
- Ingresamos en la **PESTAÑA: PODs** & se mostrarán los **PODs** asociados al **SOFTWARE** instalado en este **NAMESPACE**:
  - ✓ **OPERADORES: ACE, MQ.**
  - ✓ **APLICACIONES: ACE, DASHBOARD, MQ.**

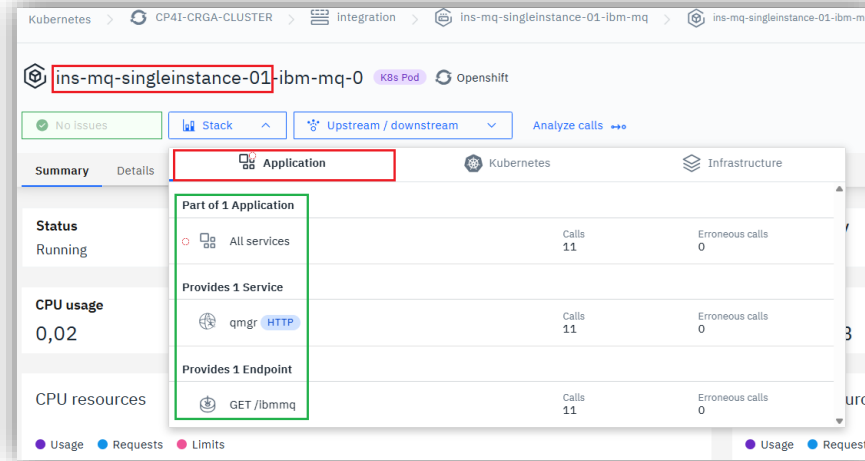
**IMPORTANTE:** "Se debe identificar el nombre en este caso del **POD** asociado a **IBM MQ**".



- Luego, se mostrará un **DASHBOARD** asociado a la **INSTANCIA** de **MQM**: **ins-mq-singleinstance-01**...
- Aquí se podría ver información asociado a los **RECURSOS** dentro del **POD** del **MQM**.



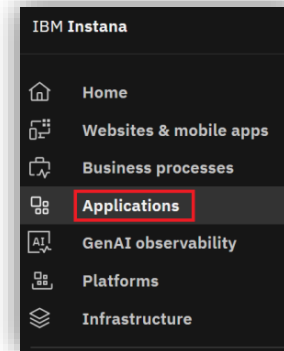
- Luego, ingresar dentro del **POD** asociado al **MQM** en este caso: **ints-mq-singleinstance-01-xxxxx**.
- Al seleccionar en **STACK** aparecerán las opciones: **APPLICATION, KUBERNETES, INFRAESTRUCTURE**.
- Después, ingresar en: **STACK / APPLICATION** & se mostrarán el **SERVICIO** de **MQ** identificado.



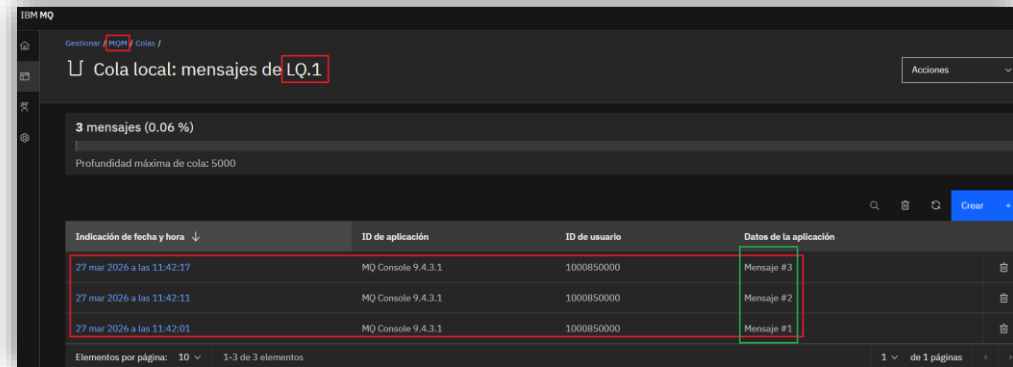
- Otra opción es seleccionar: **STACK / KUBERNETES** & se mostrarán los **RECURSOS** de **OPENSIFT** asociados al **SERVICIO** de **MQ**.

- Otra opción es seleccionar: **STACK / INFRASTRUCTURE** & se mostrarán los **RECURSOS** de **INFRA** asociados al **SERVICIO** de **MQ**.

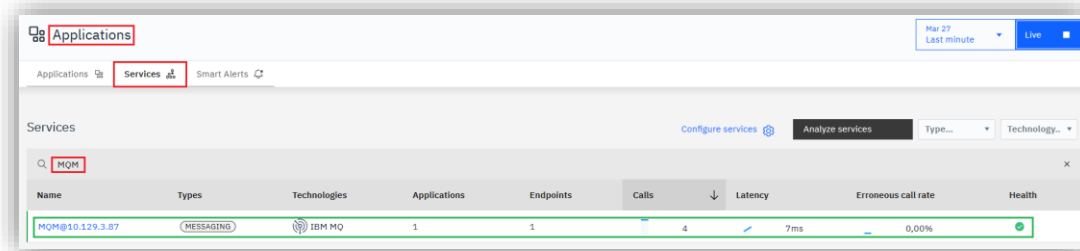
- Luego, del **MENÚ PRINCIPAL** se procede a seleccionar: **APPLICATIONS**.



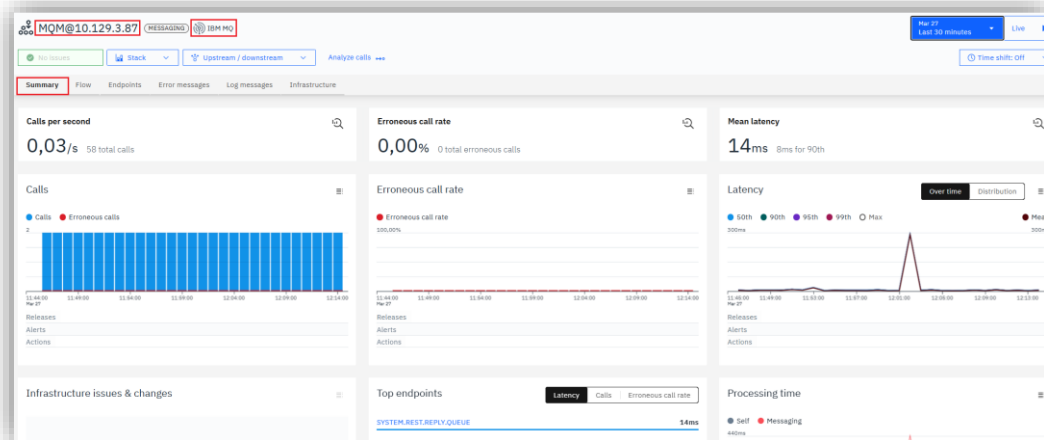
- Luego, en **PARALELO** se procede a ingresar a la **CONSOLA WEB** del **MQM**.
- Aquí se selecciona la **QUEUE** en este caso: **LQ.1** & se le registran **3 MENSAJES** para las pruebas como se muestra en la **IMAGEN**.



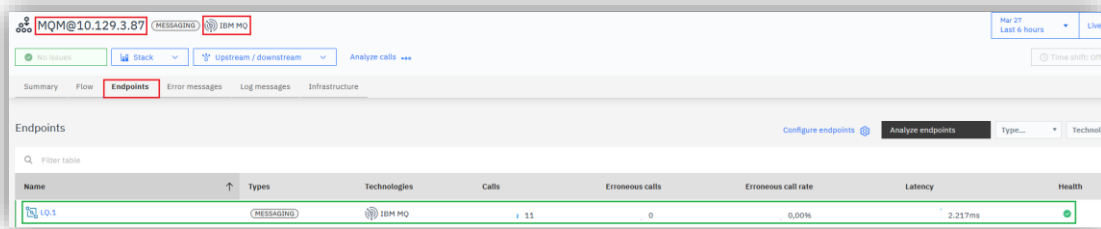
- Luego, se procede a seleccionar la **PESTAÑA: SERVICES** & ingresar en el filtro el nombre del **MQM (que figura en la CONSOLA)**.
- En este caso: **MQM**.
- Después, se procederá a mostrar el resultado de la **TECNOLOGÍA** asociada al **SERVICIO** & el resultado en caliente de los **REQUEST** recibidos.



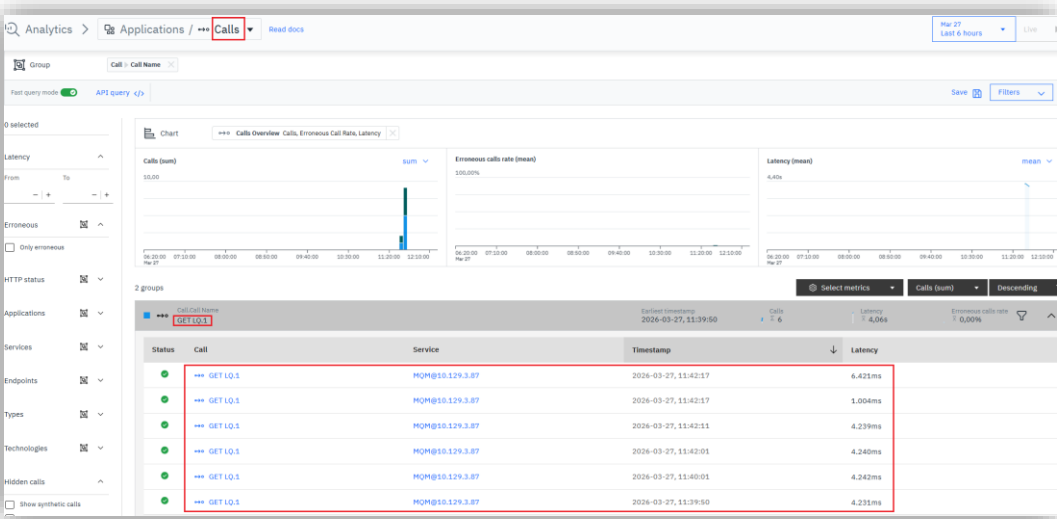
- Luego, ingresando en el **SERVICE** se muestra un **DASHBOARD** completo de **INDICADORES**.
- Estos **INDICADORES** están asociados a las **TRAZAS** de los registros ejecutados contra el **MQ**.



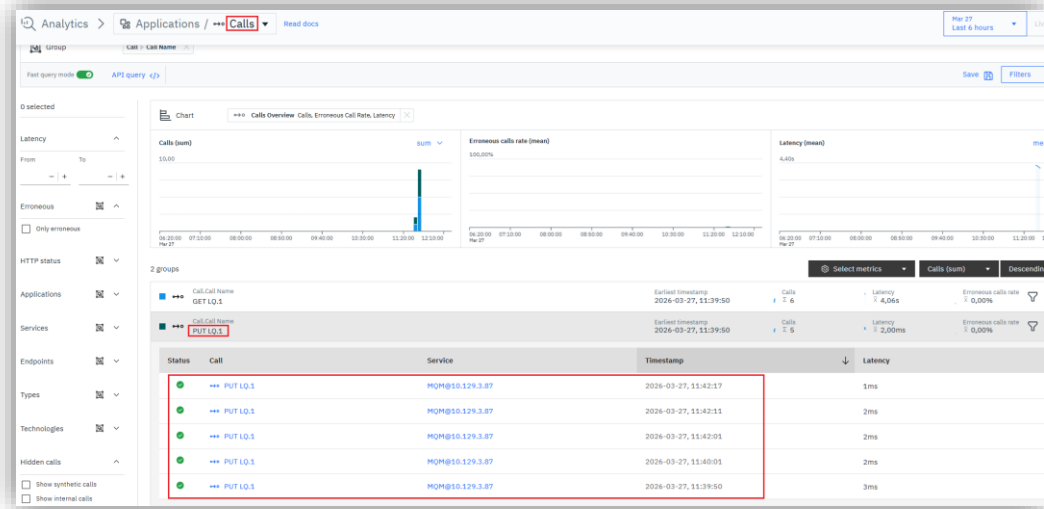
- Luego, se procede seleccionar en la **PESTAÑA: ENDPOINTS**.
- Aquí se carga en este caso el **ENDPOINT** asociado al nombre de **QUEUE** que se está trabajando: **LQ.1**.
- Después, se procede a ingresar en dicho **ENDPOINT**.



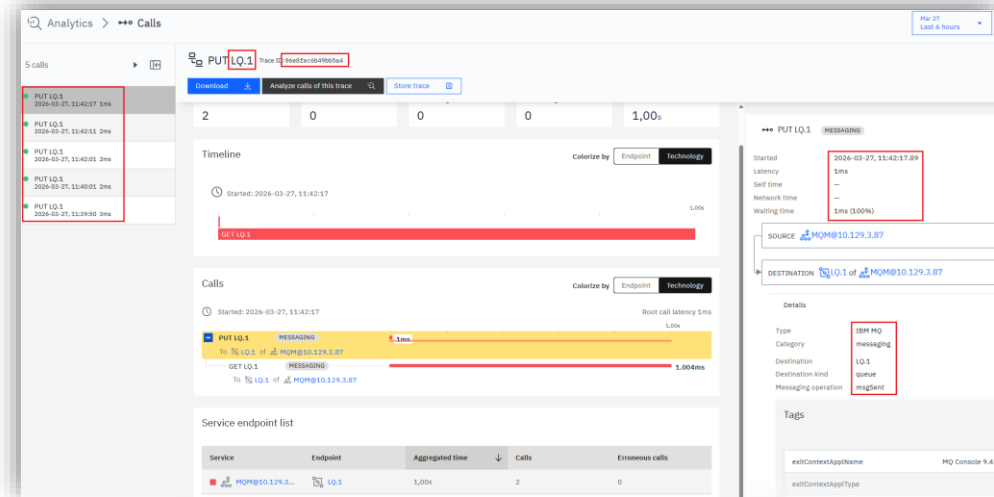
- Luego, estando dentro del **ENDPOINT** se selecciona en **CALLS** & dentro se cargarán las diferentes **INTERACCIONES** que ha habido con la **QUEUE**, en este caso de tipo: **GET**.



- Luego, así mismo se puede seleccionar las diferentes **INTERACCIONES** que ha habido con la **QUEUE**, en este caso de tipo: **POST**.



- Finalmente, ingresando en cualquiera de estas, se apertura un **DASHBOARD** con todo el **DETALLE** asociado al **MENSAJE** dentro de dicha **QUEUE**: **LQ.1 (ENDPOINT)**.

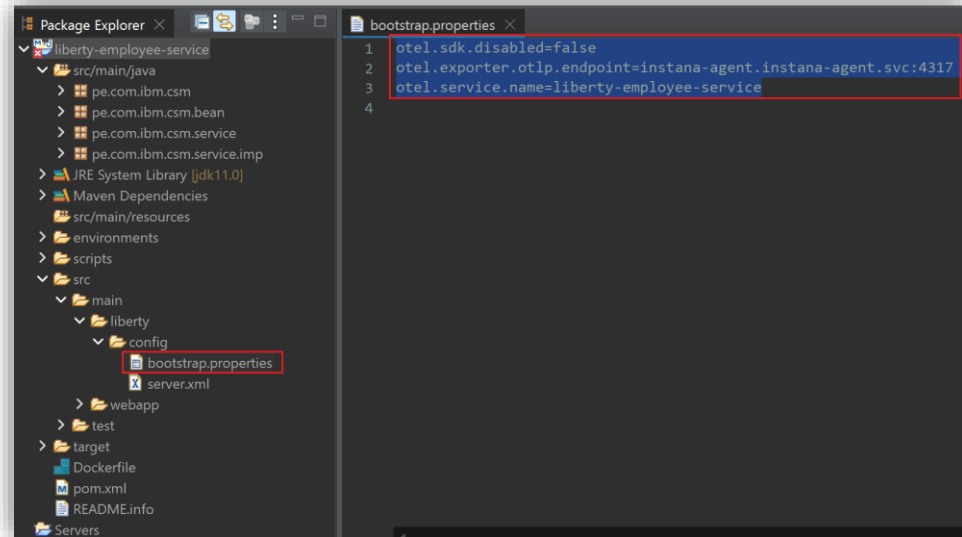


## V. IBM LIBERTY: (CONFIGURACIÓN)

▪ Se procede a validar la **CONFIGURACIÓN** existente en para el soporte de: **OPEN-TELEMETRY**.

▪ En este caso la **CONFIGURACIÓN** sería:

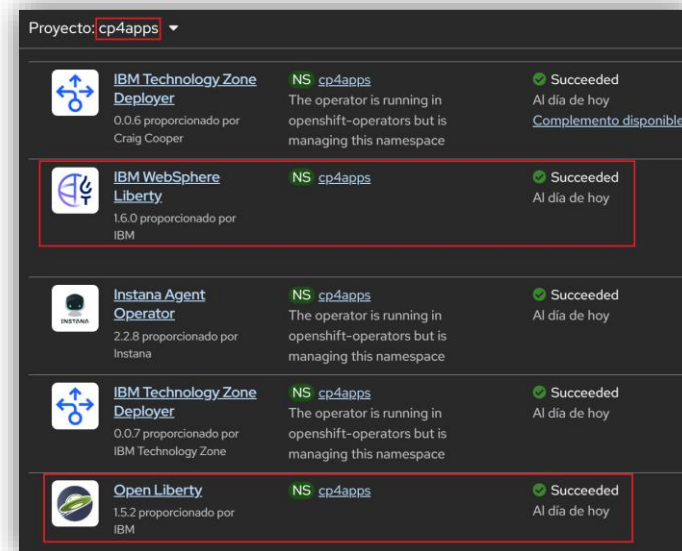
- ✓ `otel.sdk.disabled=false`
- ✓ `otel.exporter.otlp.endpoint=instana-agent.instana-agent.svc:4317`
- ✓ `otel.service.name=liberty-employee-service`



▪ Luego, se valida que **ESCENARIO** de implementación a nivel de **OPERATORS** se está trabajando con **LIBERTY**:

- ✓ **IBM WEBSHERE LIBERTY.**
- ✓ **IBM OPEN LIBERTY.**

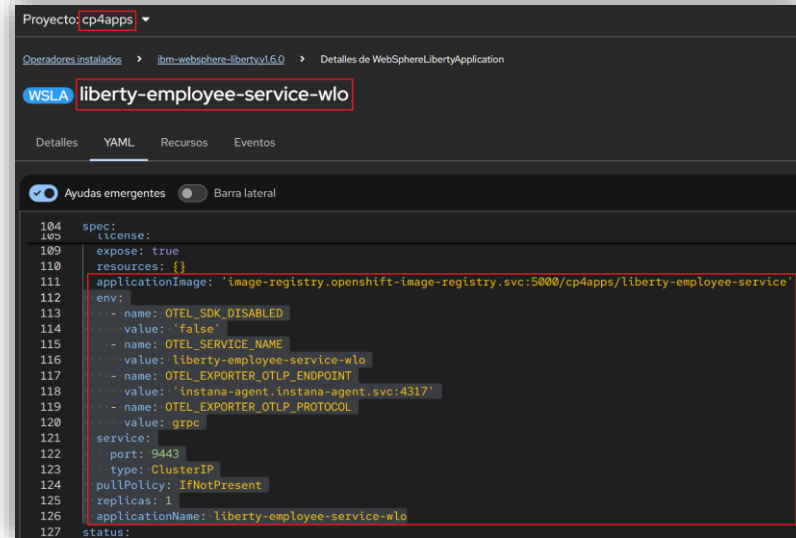
**IMPORTANTE:** "En este caso se mostrará como trabajar con ambos **ESCENARIOS**".



▪ Luego, ingresar en ambas **INSTANCIAS** de **LIBERTY** creadas & validar que existan las **PROPIEDADES**:

- ✓ **APPLICATION IMAGE.**
- ✓ **APPLICATION NAME.**
- ✓ **VARIABLES DE ENTORNO.**

**IMPORTANTE:** “En este caso el **IMAGE-STREAM** del **MICROSERVICIO** es reutilizada para ambas **INSTANCIAS (WLO & OLO)**”.



Proyecto: cp4apps

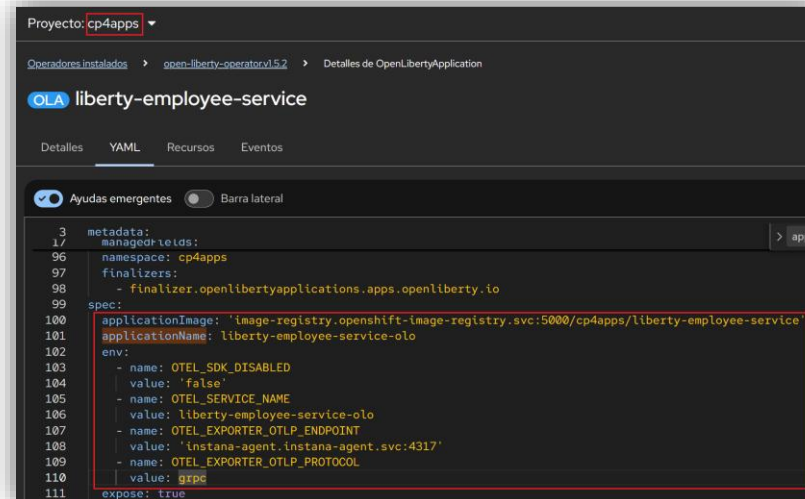
Operadores instalados > ibm-websphere-libertyv1.6.0 > Detalles de WebSphereLibertyApplication

WSLA liberty-employee-service-wlo

Detalles YAML Recursos Eventos

Ayudas emergentes Barra lateral

```
104 spec:
105   license:
106   expose: true
107   resources: {}
108   applicationImage: 'image-registry.openshift-image-registry.svc:5000/cp4apps/liberty-employee-service'
109   env:
110     - name: OTEL_SDK_DISABLED
111       value: 'false'
112     - name: OTEL_SERVICE_NAME
113       value: liberty-employee-service-wlo
114     - name: OTEL_EXPORTER_OTLP_ENDPOINT
115       value: 'instana-agent.instana-agent.svc:4317'
116     - name: OTEL_EXPORTER_OTLP_PROTOCOL
117       value: grpc
118   service:
119     port: 9443
120     type: ClusterIP
121     pullPolicy: IfNotPresent
122     replicas: 1
123     applicationName: liberty-employee-service-wlo
124   status:
```



Proyecto: cp4apps

Operadores instalados > open-liberty-operatorv1.5.2 > Detalles de OpenLibertyApplication

OLA liberty-employee-service

Detalles YAML Recursos Eventos

Ayudas emergentes Barra lateral

```
3 metadata:
4   managedFields:
5   namespace: cp4apps
6   finalizers:
7     - finalizer.openlibertyapplications.apps.openliberty.io
8   spec:
9     applicationImage: 'image-registry.openshift-image-registry.svc:5000/cp4apps/liberty-employee-service'
10    applicationName: liberty-employee-service-olo
11    env:
12      - name: OTEL_SDK_DISABLED
13        value: 'false'
14      - name: OTEL_SERVICE_NAME
15        value: liberty-employee-service-olo
16      - name: OTEL_EXPORTER_OTLP_ENDPOINT
17        value: 'instana-agent.instana-agent.svc:4317'
18      - name: OTEL_EXPORTER_OTLP_PROTOCOL
19        value: grpc
20    expose: true
```

- Luego, antes la **MODIFICACIÓN** de las **2 INSTANCIAS** de **LIBERTY**, validar el **REINICIO** de los **PODS** asociados.

**IMPORTANTE:** "Si no se reinician, hay que forzar **MANUALMENTE** el **REINICIO**".

Nombre	Estado	Listo	Se reinicia	Propietario	Memoria	CPU	Creado
liberty-employee-service-7c76f99964-6nzkd	Running	1/1	0	liberty-employee-service-7c76f99964	241,9 MIB	0,093 núcleos	hace 1 minuto
liberty-employee-service-wlo-59c76f85b5-fn6wf	Running	1/1	0	liberty-employee-service-wlo-59c76f85b5	232,7 MIB	0,082 núcleos	hace 1 minuto
olo-controller-manager-6868c9f56c-ozsn7	Running	1/1	0	olo-controller-manager-6868c9f56c	30,5 MIB	0,002 núcleos	27 mar. 2026, 10:15 a. m.
wlo-controller-manager-5d8c8889d6-fgqg5	Running	1/1	0	wlo-controller-manager-5d8c8889d6	34,2 MIB	0,002 núcleos	28 mar. 2026, 6:46 p. m.

- Luego, se procede a lanzar **REQUEST** contra las **2 URLs**, asociadas a las **2 INSTANCIAS** de **LIBERTY** trabajadas:

- En este caso desplegadas en **OPENSIFT**:

- ✓ **WLO:** <https://liberty-employee-service-wlo-cp4apps-new.apps.itz-jivqca.infra01-lb.dal14.techzone.ibm.com/liberty-employee-service/ibm/employeeService/getEmpleados>
- ✓ **OLO:** <https://liberty-employee-service-cp4apps-new.apps.itz-jivqca.infra01-lb.dal14.techzone.ibm.com/liberty-employee-service/ibm/employeeService/getEmpleados>

```

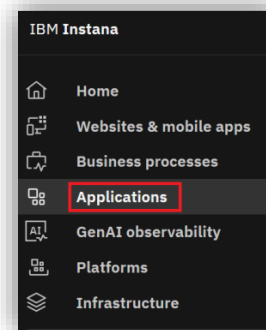
GET https://liberty-employee-service-wlo-cp4apps-new.apps.itz-jivqca.infra01-lb.dal14.techzone.ibm.com/liberty-employ-
Body
200 OK · 154 ms · 465 B
JSON
[
  {
    "apellidos": "LLANOS SILVA",
    "dni": 42814133,
    "edad": 30,
    "id": 1,
    "nombres": "CESAR ARTURO"
  },
  {
    "apellidos": "FIGO TUDELA",
    "dni": 41816133,
    "edad": 43,
    "id": 2,
    "nombres": "LUIS ARMANDO"
  },
  {
    "apellidos": "MUELA MALDONADO",
    "dni": 35561313,
    "edad": 44,
    "id": 3,
    "nombres": "IVAN ARTURO"
  }
]

```

```
GET https://liberty-employee-service-cp4apps-new.apps.itz-ijvqca.infra01-lb.dal14.techzone.ibm.com/liberty-employee-s

Body
JSON
[
  {
    "apellidos": "LLANOS SILVA",
    "dni": 42814133,
    "edad": 30,
    "id": 1,
    "nombres": "CESAR ARTURO"
  },
  {
    "apellidos": "FIGO TUDELA",
    "dni": 41816133,
    "edad": 43,
    "id": 2,
    "nombres": "LUIS ARMANDO"
  },
  {
    "apellidos": "MUELA MALDONADO",
    "dni": 95861313,
    "edad": 44,
    "id": 3,
    "nombres": "IVAN ARTURO"
  }
]
```

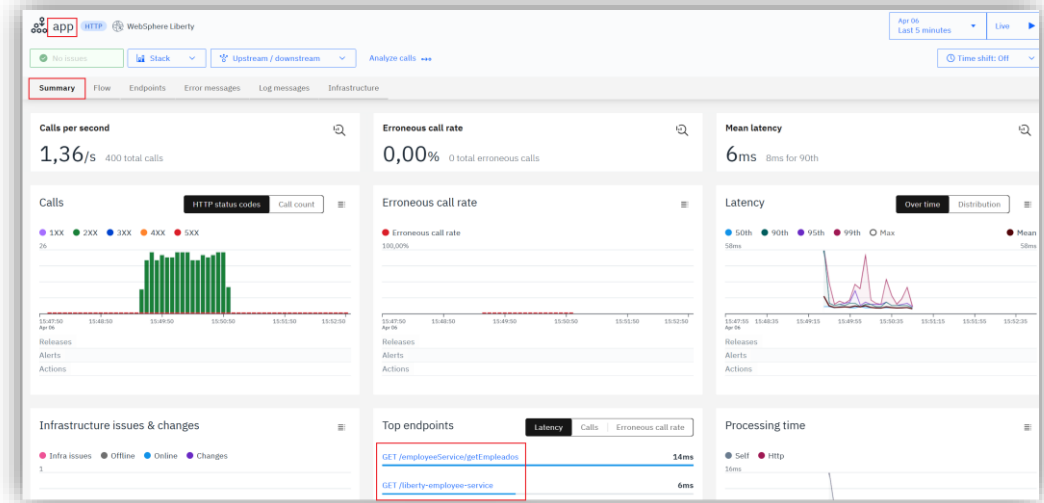
- Luego, del **MENÚ PRINCIPAL** en **INSTANA** se procede a seleccionar: **APPLICATIONS**.



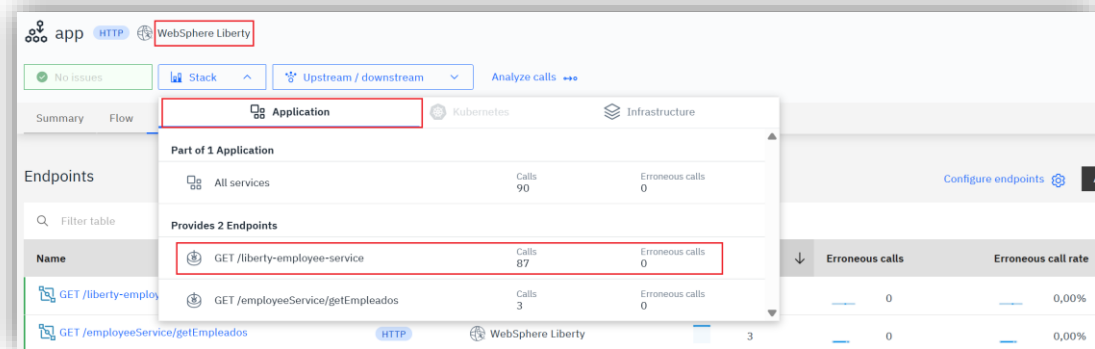
- Luego, se procede a seleccionar la **PESTAÑA: SERVICES** e ingresar un filtro asociado a **LIBERTY**.
- En este caso: **app**.
- Se procederá a mostrar el resultado de la **TECNOLOGÍA** asociada al **SERVICIO** & el resultado en caliente de los **REQUEST** obtenidos.

Name	Types	Technologies	Applications	Endpoints	Calls	Latency	Erroneous call rate	Health
app	HTTP	WebSphere Liberty	1	2	147	7ms	0.00%	🟢

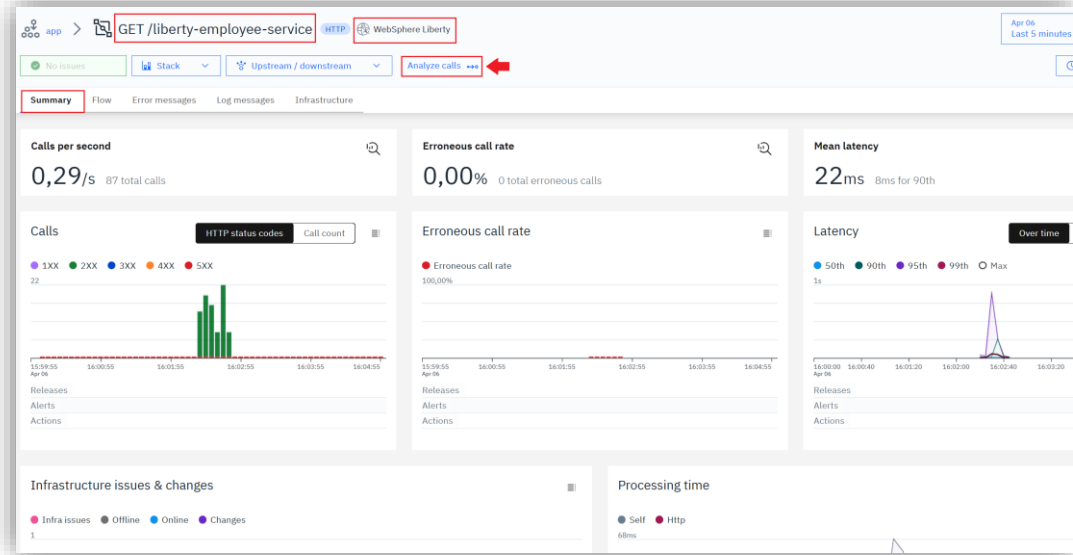
- Luego, ingresando en el **SERVICE** se muestra un **DASHBOARD** completo de **INDICADORES**.
- Estos **INDICADORES** están asociados a las **TRAZAS** de los registros ejecutados contra el **LIBERTY**.



- Luego, al seleccionar en **STACK** & aparecerán las opciones: **APPLICATION**, **KUBERNETES**, **INFRAESTRUCTURE**.
- Aquí se procede a seleccionar: **STACK / APPLICATION** & se mostrarán los **ENDPOINT** asociados a: **LIBERTY**.



- Luego, se procede a seleccionar en **CALLS** & se cargarán las diferentes **INTERACCIONES** que ha habido contra **LIBERTY**, en este caso de tipo: **GET**.



Analytics > Applications / ++ Calls

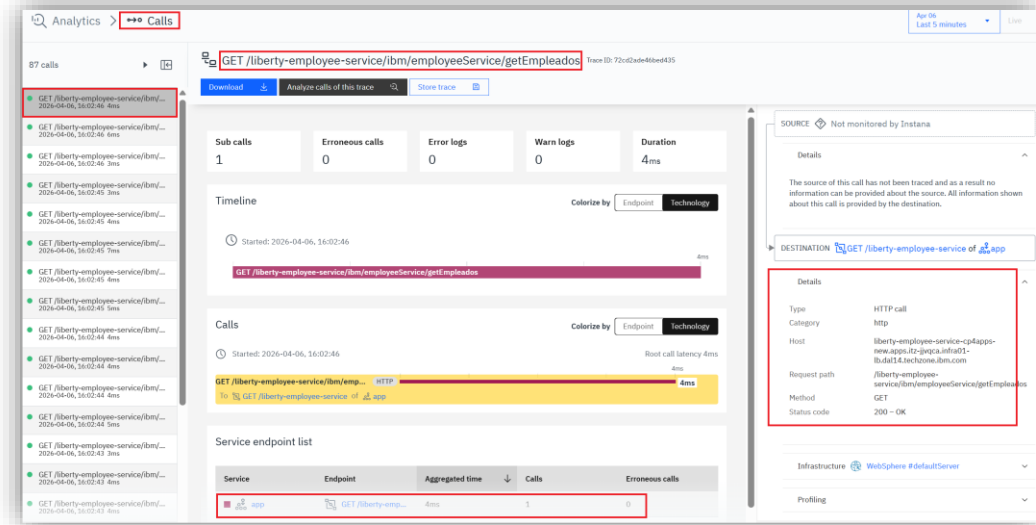
Latency (mean): 68ms

Erroneous calls rate (mean): 100,00%

1 groups

Status	Call	Service	Timestamp	Latency
✓	+++ GET /liberty-employee-service/bn/employee...	app	2026-04-06, 16:02:46	4ms
✓	+++ GET /liberty-employee-service/bn/employee...	app	2026-04-06, 16:02:46	6ms
✓	+++ GET /liberty-employee-service/bn/employee...	app	2026-04-06, 16:02:46	3ms
✓	+++ GET /liberty-employee-service/bn/employee...	app	2026-04-06, 16:02:45	3ms
✓	+++ GET /liberty-employee-service/bn/employee...	app	2026-04-06, 16:02:45	4ms
✓	+++ GET /liberty-employee-service/bn/employee...	app	2026-04-06, 16:02:45	7ms
✓	+++ GET /liberty-employee-service/bn/employee...	app	2026-04-06, 16:02:45	4ms
✓	+++ GET /liberty-employee-service/bn/employee...	app	2026-04-06, 16:02:45	5ms

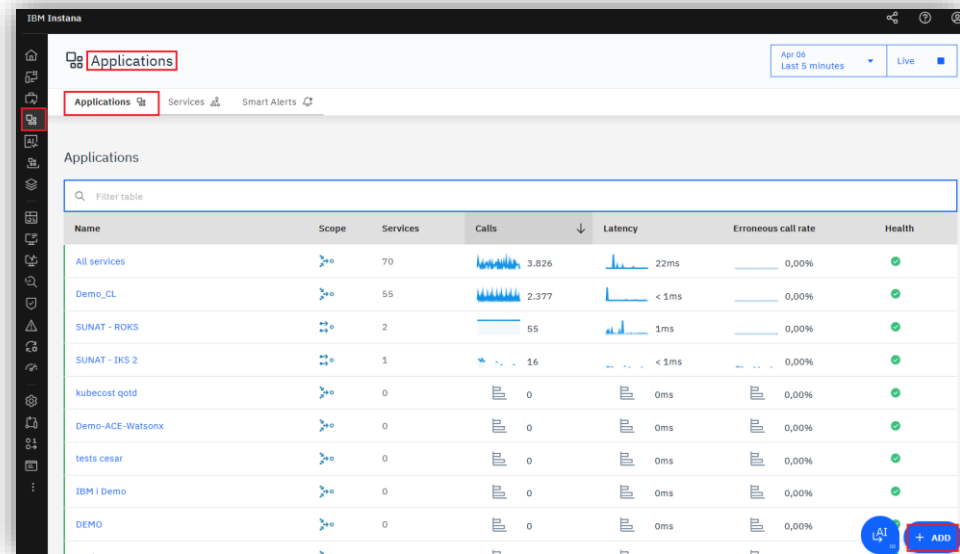
- Finalmente, se muestra el listado de **REQUEST** procesados por el **SERVICIO** del **LIBERTY**.
- Así mismo, al seleccionar cualquiera se procede a obtener todo el **DETALLE** del mensaje.



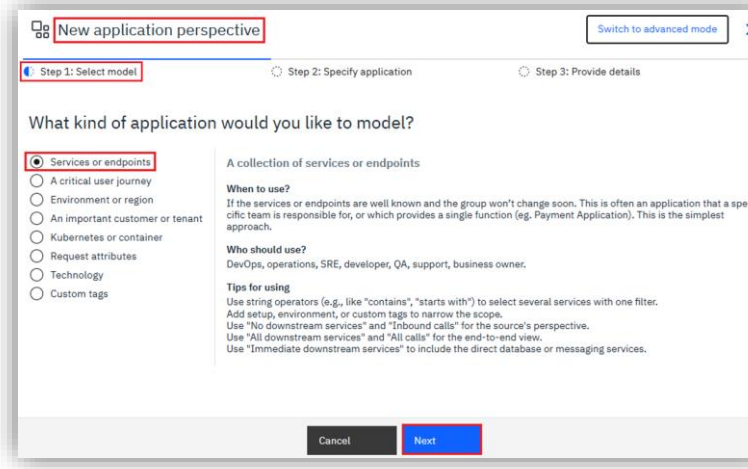
## VI. ALERT MANAGER: (CONFIGURACIÓN)

### A. EVENTS (CONFIGURACIÓN):

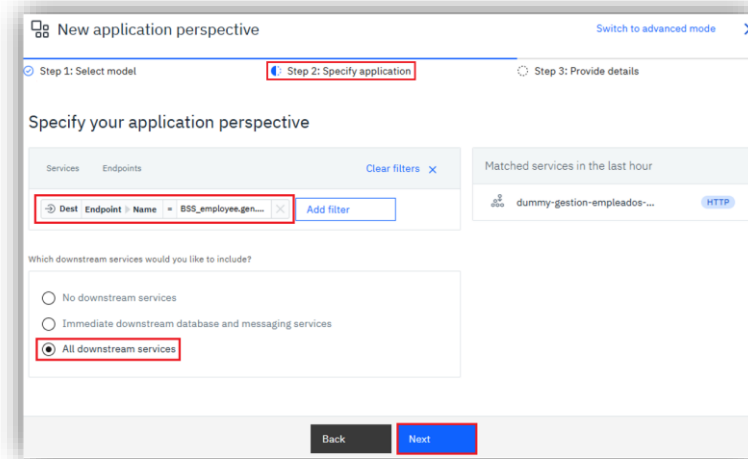
- Para la **CONFIGURACIÓN & FUNCIONAMIENTO** requerido en la creación de un **EVENT**, es necesaria la existencia de un **APPLICATION**.
- Luego, desde el **MENÚ PRINCIPAL** seleccionar: **APPLICATIONS**, en la pestaña seleccionar: **APPLICATIONS** & dar en **ADD**.



- Luego, se mostrará un **POPUP** para la configuración.
- **PASO#1:** Esta **CONFIGURACIÓN** se asocia al tipo de **APLICACIÓN** deseada:
  - ✓ **SERVICES OR ENDPOINTS.**



- **PASO#2:** aquí se debe ingresar el **FILTRO** deseado (*por medio de que se realizará*).
- El objetivo en mi caso es que el **APPLICATION** por detrás se conecte al **ENDPOINT** que usan las **CAPACIDADES** de **INTEGRACIÓN** que estoy observando en **INSTANA: ACE, MQ & LIBERTY**.
- Debido a ello, se relacionará con el **ENDPOINT** respectivo (*obtenido de: INSTANA*).
  - ✓ **IBM APP CONNECT ENTERPRISE:**  
`Endpoint/Name = BSS_employee.gen.BSS_employee`
  - ✓ **IBM MQ ADVANCE:**  
`Endpoint/Name = LQ.1`
  - ✓ **IBM LIBERTY:**  
`Endpoint/Name = GET /liberty-employee-service`



- Así mismo, seleccionar:
  - ✓ **ALL DOWNSTREAM SERVICES**

- **PASO#3:** aquí se debe ingresar el **NOMBRE** del **APPLICATION**, como se encontrará en **INSTANA**.

- Configurar en este caso:

- ✓ **NOMBRE:** **dummy-gestion-empleados-app**
- ✓ **CALLS:** **ALL CALLS**.

New application perspective

Step 1: Select model Step 2: Specify application Step 3: Provide details

What is the name of this application perspective?

Application perspective name

dummy-gestion-empleados-app

Matched services in the last hour

dummy-gestion-empleados... HTTP

Are you interested just in the calls to this application, or also the internal calls?

Inbound calls

The dashboard is displaying information based only on the calls that are performed by the consumers of this application.

All calls

The dashboard is displaying information based on all calls that are performed within this application, by both consumers as well as internally.

Back Create

- Luego, desde el **MENÚ PRINCIPAL** ingresar otra vez en: **APPLICATIONS**, & seleccionar la **PESTAÑA: APPLICATIONS**.

- Aquí se apreciará que todos los **REQUEST** asociados en este caso al **ENDPOINT** de **ACE**, serán capturados desde el **APPLICATIONS**.

**IMPORTANTE:** "Hay que recordar que esta configuración de **APPLICATIONS**, se debe realizar en este caso para: **ACE, MQ & LIBERTY**".

Applications

Apr 06 Last minute Live

Applications

dummy-gestion-empleados

Name	Scope	Services	Calls	Latency	Erroneous call rate	Health
dummy-gestion-empleados-app		1	40	1ms	0,00%	✓

Así mismo, es **IMPORTANTE** considerar que esta **CONFIGURACIÓN**, asociada al **ENDPOINT**, para la creación de **APPLICATIONS**, es aplicable para las demás **CAPACIDADES**:

✓ **IBM APP CONNECT ENTERPRISE:**  
[dummy-gestion-empleados-app](#)

✓ **IBM MQ ADVANCE:**  
[PRONACA-app](#)

✓ **IBM LIBERTY:**  
[liberty-app](#)

The screenshot shows the 'New application perspective' wizard in three steps. Step 2, 'Specify application', shows a filter for 'Dest Endpoint Name = 104' and a list of 'Matched services in the last hour' including 'PRONACA@10.129.3.96'. Step 3, 'Provide details', shows the application perspective name 'PRONACA-app' and the selection of 'All calls' for the dashboard scope. Below the wizard is the 'Applications' dashboard showing a table with one entry: 'PRONACA-app' with 1 service, 3 calls, and a latency of < 1ms.

Name	Scope	Services	Calls	Latency	Erroneous call rate	Health
PRONACA-app	>=	1	3	< 1ms	0,00%	🟢

The screenshot shows the 'New application perspective' wizard for 'liberty-app'. Step 2 shows a filter for 'Dest Endpoint Name = GET/liberty-emplo...' and a list of 'Matched services in the last hour' including 'liberty-app'. Step 3 shows the application perspective name 'liberty-app' and the selection of 'All calls' for the dashboard scope. Below the wizard is the 'Applications' dashboard showing a table with one entry: 'liberty-app' with 1 service, 4 calls, and a latency of 25ms.

Name	Scope	Services	Calls	Latency	Erroneous call rate	Health
liberty-app	>=	1	4	25ms	0,00%	🟢

- Luego, se procede a seleccionar en: **MENÚ / SETTINGS / EVENTS**:
- El objetivo es especificar un posible **EVENTO**, para luego **CONTROLARLO** por una **ALERTA**.
- Los parámetros para configurar para el **EVENTO** son:

“**CASO**: Me gustaría recibir una **ALERTA** que me informe cuando el **POD** asociado al **MICROSERVICIO** haya excedido el **60%** de la **CPU** para la cantidad de solicitudes (**REQUEST**) que se está recibiendo”.

✓ **NOMBRE:** **IBM-EVENT-CRGA-01**

✓ **DESCRIPCIÓN:** **EVENTO QUE ALERTARÁ SOBRE UN POD, ASOCIADO A UN MICROSERVICIO INACTIVO.**

✓ **SEVERIDAD:** *Critical*

✓ **INDICENT:** *true*

✓ **GRACE PERIOD:** *60 seg.*

✓ **SOURCE:** *Build Metrics*

✓ **ENTITY-TYPE:** *Instana Agent*

✓ **METRIC:** *CPU >Load*

✓ **TIME WINDOW:** *60 seg.*

✓ **AGGREGATION:** *Max*

✓ **OPERATOR:** *>=*

✓ **COUNT:** *60*

✓ **SCOPE:** *Application Perspective*

✓ **APLICACION PERSPECTIVE:**

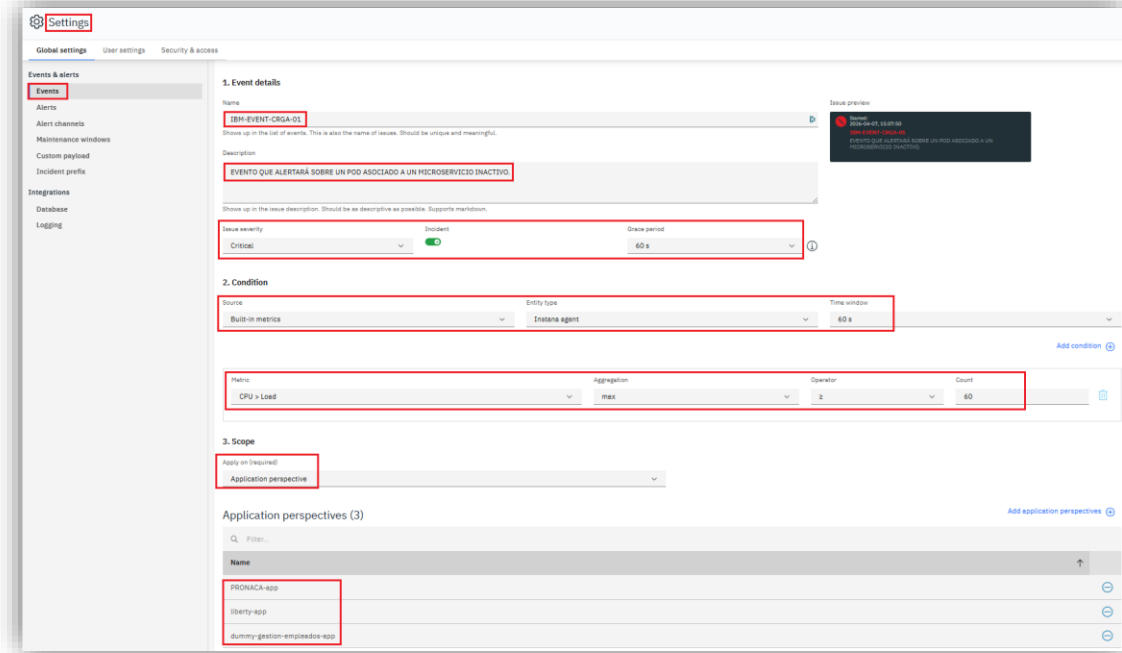
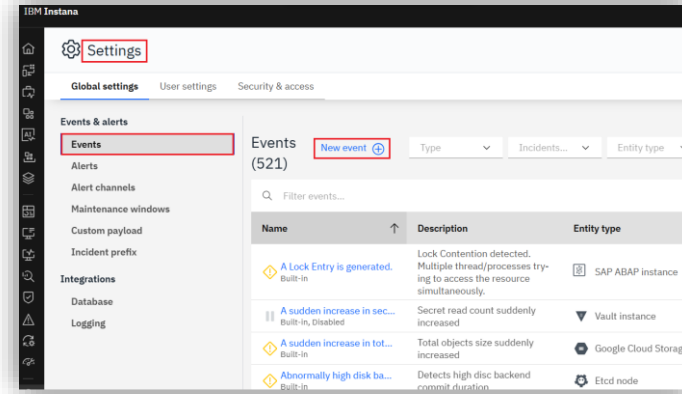
*dummy-gestion-empleados-app*

*PRONACA-app*

*liberty-app*

- Esta última parte es donde se asocia el **EVENT** con el **APPLICATION** previamente creado, en este caso asociado al: **ACE, MQ & LIBERTY**.

**IMPORTANTE:** “Siempre se debe proceder a de crear un **EVENTO** antes de una **ALERTA**”.



#### 4. Transient Events (optional)

When transient events are enabled, Instana predicts whether an event will likely resolve itself within a user-defined threshold. This prediction is based on historical event durations for the same configuration and entity.  
Transient events can be hidden on the events page using the DFO: "event.isTransient:false".

Enable

Threshold

5 - + Minutes

Recommended threshold is five minutes.

Notification

Send an alert immediately.

Only send an alert if an event persists after the threshold.

Cancel Save

- Finalmente, se aprecia el **EVENT** creado filtrando por el nombre: **IBM-EVENT-CRGA-01**.

Settings

Global settings User settings Security & access

Events & alerts

Events Alerts Alert channels Maintenance windows Custom payload Incident prefix Integrations Database

Events (1/522)

New event

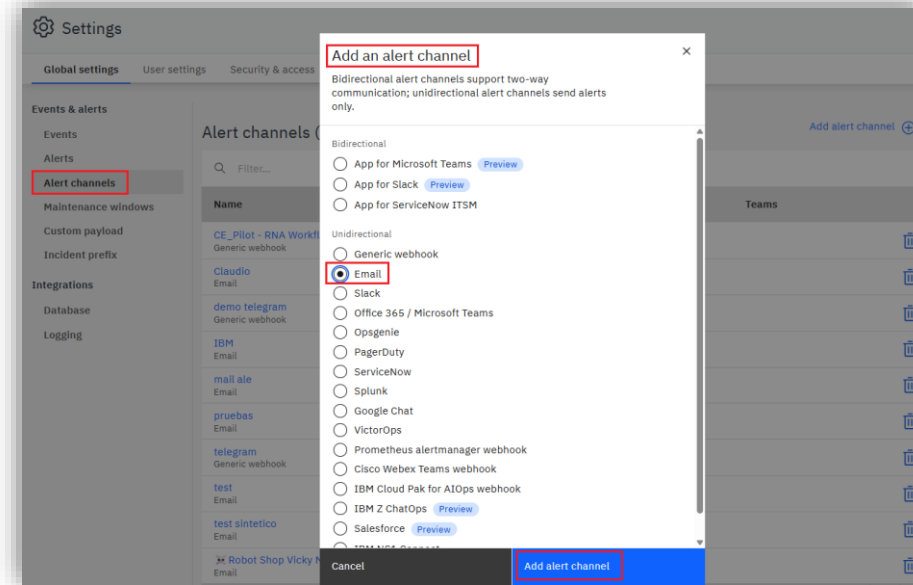
Type Incidents & severit... Entity type State...

CRGA

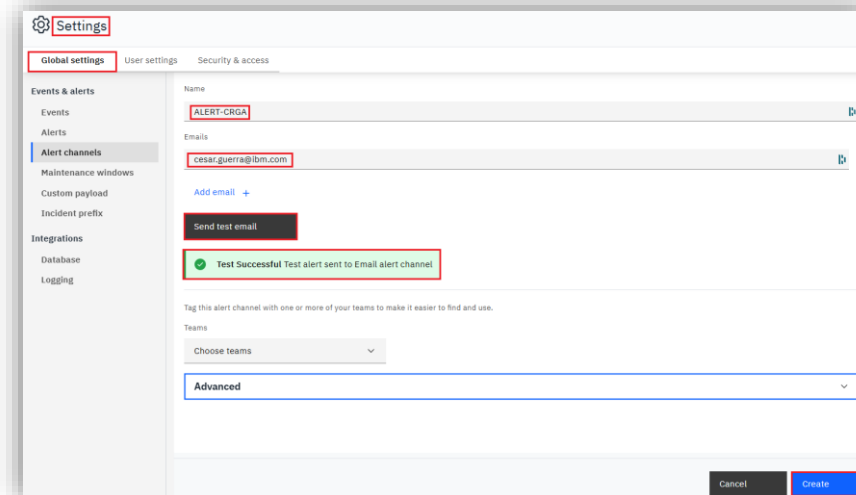
Name	Description	Entity type
IBM-EVENT-CRGA-01	EVENTO QUE ALERTARÁ SOBRE UN POD ASOCIADO A UN MICROSERVICIO INACTIVO.	Instana agent

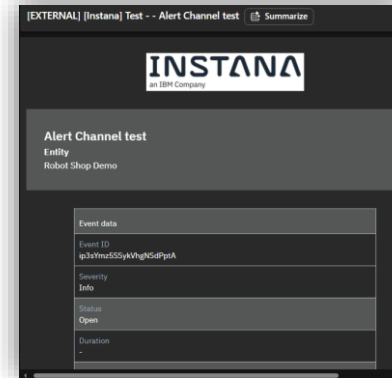
## B. ALERTs (CONFIGURACIÓN):

- Habiendo ya creado el **EVENT** se procede a hacer las configuraciones para la creación del **ALERT**.
- Se procede a seleccionar en: **SETTINGS** / **ALERT CHANNELS** & dar en: **ADD ALERT CHANNEL**.
- Aquí el objetivo es crear un **DESTINO** específico & reutilizable, para la **ALERTA**, en este caso se selecciona **EMAIL**.

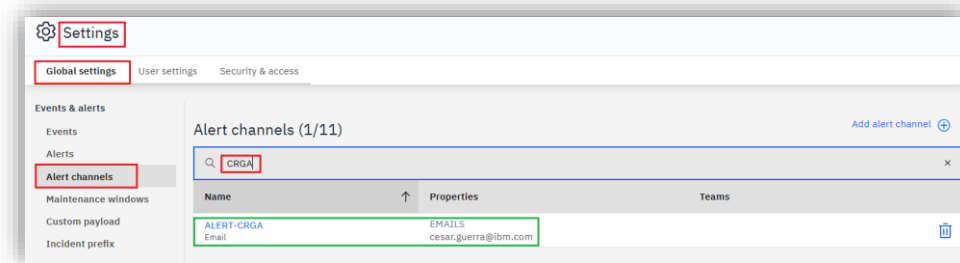


- Luego, se procede a ingresar los **DATOS** para la **CONFIGURACIÓN**:
  - ✓ **NOMBRE:** **ALERT-CRGA**
  - ✓ **EMAIL:** **cesar.guerra@ibm.com**
- Así mismo, realizar una prueba en envío de **EMAIL**, seleccionando el botón: **SEND TEST EMAIL**.
- Se requiere validar que el **EMAIL** similar a la foto, logre llegar al **CORREO** registrado.





- Luego, se aprecia el **ALERT-CHANNEL** creado filtrando por el nombre: **ALERT-CRGA**.



▪ Luego, se procede a seleccionar: **MENÚ / SETTINGS / ALERTS**:

▪ Aquí el objetivo es crear una **ALERTA** en **INSTANA** & configurar sus **PROPIEDADES**, para que funcione en diferentes **ESCENARIOS**:

▪ En este caso la **CONFIGURACIÓN** es:

✓ **NAME:** **Evento-CRGA**

✓ **EVENTS:** **Alert on Event Types**

✓ **EVENT-TYPE:**

- **Incidents:** *True*
- **Critical Issues:** *True*
- **Warning Issues:** *True*
- **Changes:** *False*
- **Online:** *False*
- **Offline:** *False*
- **Monitoring Issues:** *True*

✓ **SCOPE:** **Application Perspective**

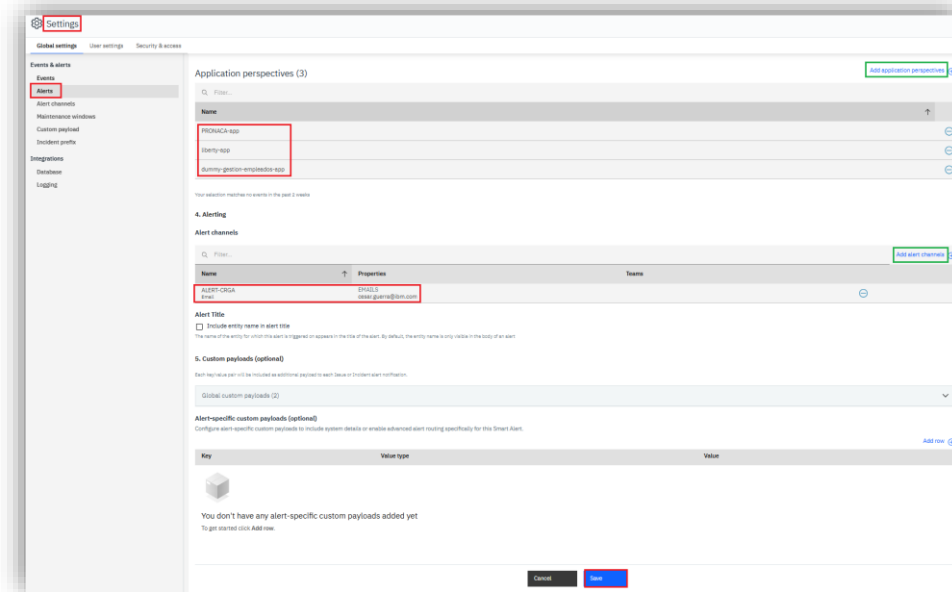
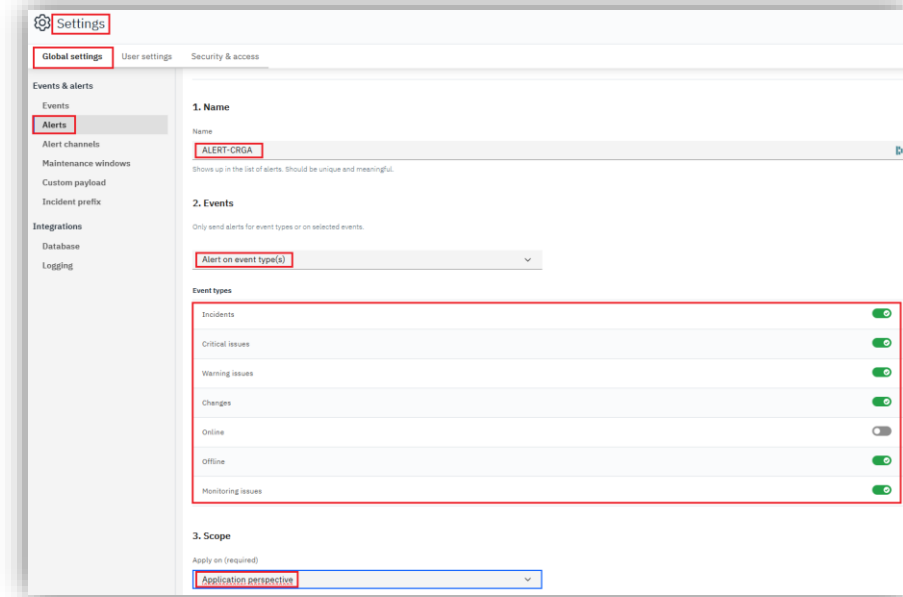
✓ **APPLICATION PERSPECTIVE:**

**dummy-gestion-empleados-app**  
**PRONACA-app**  
**liberty-app**

▪ Esta última parte es donde se asocia el **ALERT** con el **APPLICATION** previamente creado, en este caso asociado al: **ACE, MQ & LIBERTY**.

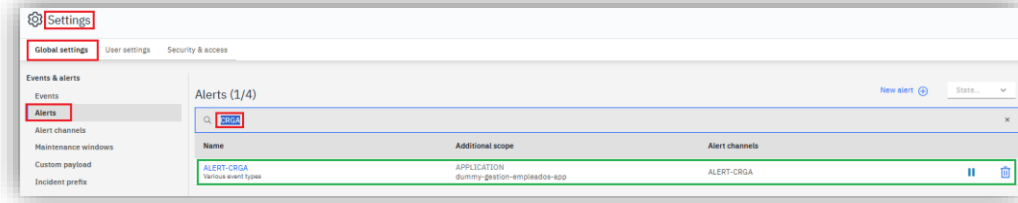
✓ **ALERT-CHANNELS:** **ALERT-CRGA**

▪ Esta última parte es donde se asocia el **ALERT** con el **ALERT CHANNEL** previamente creado.

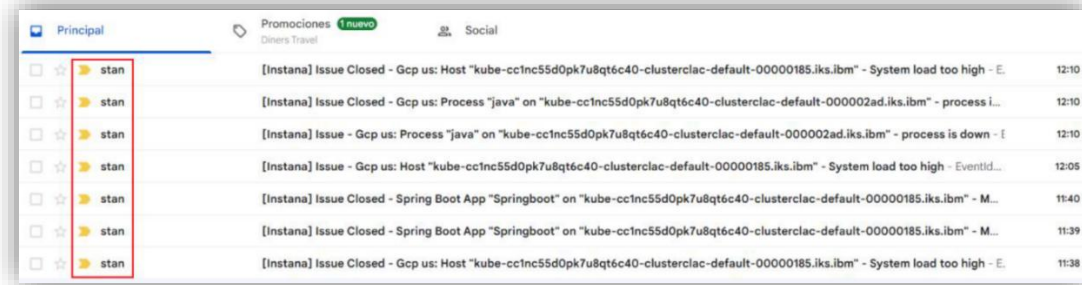


- Luego, se aprecia la **ALERT** creada filtrando por el nombre: **ALERT-CRGA**.

**IMPORTANTE:** "En este caso se ha creado tanto la **ALERT** como el **ALERT-CHANNEL**, con el mismo **NOMBRE** para rápidamente identificar la relación".



- Luego, se aprecia como deberían de llegar los **ALERTs**, en este caso por **EMAIL**, sobre **PROBLEMAS** detectados por **INSTANA**.



- Finalmente, ingresar en: **MENU / EVENTS:**
- Aquí el objetivo es **VISUALIZAR** las **INCIDENCIAS** existentes de forma **CENTRALIZADA**, en base a las **ALERTs** previamente definidas.
- La búsqueda del **EVENTO** se hace ingresando el nombre del **APPLICATION:**
  - ✓ **IBM APP CONNECT ENTERPRISE:**  
[dummy-gestion-empleados-app](#)
  - ✓ **IBM MQ ADVANCE:**  
[PRONACA-app](#)
  - ✓ **IBM LIBERTY:**  
[liberty-app](#)

