Troubleshoot Concerto Nodes

For supported software information, click here.

This article describes how to troubleshoot Versa Concerto and its various services. Concerto supports the following services:

- Apache Kafka—Distributed event-streaming platform. Concerto uses Apache Kafka is used for interservice communication and communication with Versa Director and Versa Analytics.
- Apache Solr—Scalable, distributed indexing service.
- Apache Zookeeper—Service for coordinating distributed applications.
  - Apache Kafka uses ZooKeeper to store persistent cluster metadata.
  - Patroni uses Zookeeper for leader election.
  - Concerto mgmt-service uses Zookeeper to maintain the state of the cluster.
- Docker Swarm—Container orchestration tool for managing and scheduling containers.
  - Concerto uses Docker Swarm to schedule and replicate services.
  - The Docker overlay network creates a secure distributed network for interservice communication.
  - The routing mesh enables each node in the swarm to accept connections on published ports for any service running in the swarm, even if no task is running on the node.
- Glances—Cross-platform, curses-based system-monitoring tool written in Python. Concerto uses Glances to monitor system resources, such as CPU, disk, and memory, and to raise alarms.
- GlusterFS—Scale-out, software-based, network-attached filesystem. Concerto uses GlusterFS for filesystem replication. Any file present in the /var/versa/ecp/share directory is replicated to all the nodes in the cluster.
- PostgreSQL/Patroni—Patroni is a framework for providing high availability for PostgreSQL. PostgreSQL is the main datastore for Concerto.
- Traefik—Reverse proxy and load balancer. Concerto uses Traefik as a reverse proxy for routing incoming requests from the client (web browser). Zookeeper uses Traefik as a Layer 4 load balancer.

CLI Troubleshooting Tools

This section describes the CLI commands you can use to troubleshoot Concerto.

- **vsh status**—Verify the service status.

```
admin@concerto-1:$ vsh status
postgresql          is Running
```

```
zookeeper            is Running
kafka                is Running
solr                 is Running
glances              is Running
mgmt-service         is Running
web-service          is Running
cache-service        is Running
core-service         is Running
monitoring-service   is Running
traefik              is Running
```

- **vsh cluster info**—Verify the cluster status.

```
admin@concerto-1:$ vsh cluster info

Concerto Cluster Status
-----------------------------------------------

   Node Name:             concerto-3
   IP Address:            10.40.30.80
   Operational Status:    secondary
   Configured Status:     primary
   Docker Node Status:    ready
   Node Reachability:     reachable
   GlusterFS Status:      good

   Node Name:             concerto-1
   IP Address:            10.48.7.81
   Operational Status:    primary
   Configured Status:     secondary
   Docker Node Status:    ready
   Node Reachability:     reachable
   GlusterFS Status:      good

   Node Name:             concerto-2
   IP Address:            10.48.7.82
   Operational Status:    arbiter
   Configured Status:     arbiter
   Docker Node Status:    ready
   Node Reachability:     reachable
   GlusterFS Status:      good
```

- **vsh database connect**—Connect to the PostgreSQL database shell (psql).

```
admin@concerto-1:$ vsh database connect portal
Connecting to database :  portal
User : vnms
Password for user vnms:
psql (12.5 (Debian 12.5-1.pgdg100+1), server 12.4 (Debian 12.4-1.pgdg100+1))
Type "help" for help.
portal=#
```

- **docker stack ls**—List all the Docker stacks in the cluster.

```
admin@concerto-1:$ docker stack ls
NAME                SERVICES            ORCHESTRATOR
ecp                 3                   Swarm
glances             3                   Swarm
hazelcast           1                   Swarm
kafka               6                   Swarm
misc                2                   Swarm
postgres            4                   Swarm
solr                1                   Swarm
traefik             1                   Swarm
```

- **docker stack ps** *stack-name*—Display information about a specific Docker stack.

```
admin@concerto-1:$ docker stack ps --no-trunc ecp
ID                  NAME                IMAGE                                                NODE                DESIRED
STATE       CURRENT STATE           ERROR               PORTS
gklzpq8bezs7        ecp_core-service.1        artifacts.versa-networks.com:8443/core-service:latest
concerto-1          Running             Running 2 minutes ago
rutw3e6wnerc        ecp_web-service.1         artifacts.versa-networks.com:8443/web-service:latest
concerto-1          Running             Running 2 minutes ago
q17hpiwd8ap8        ecp_monitoring-service.1  artifacts.versa-networks.com:8443/monitoring-service:latest
concerto-1          Running             Running 2 minutes ago
```

- **docker service ls**—List all the Docker services in the cluster.

```
admin@concerto-1:$ docker service ls

ID                  NAME                MODE                REPLICAS
IMAGE                                          PORTS
nvpe2hoppp6q        ecp_core-service          replicated          1/1                 artifacts.versa-networks.
com:8443/core-service:latest
rso7f1xfc4pe        ecp_monitoring-service    replicated          1/1                 artifacts.versa-networks.
com:8443/monitoring-service:latest
jvhtglgjyrpc        ecp_web-service           replicated          1/1                 artifacts.versa-networks.com:8443/
web-service:latest
vm7h6chg4wwv        glances_system-service1   replicated          1/1                 artifacts.versa-networks.
com:8443/glances:latest-alpine
yu5juld3jzjj        glances_system-service2   replicated          1/1                 artifacts.versa-networks.
com:8443/glances:latest-alpine
9cfcn4ox0xko        glances_system-service3   replicated          1/1                 artifacts.versa-networks.
com:8443/glances:latest-alpine
r0761cnj7isa        hazelcast_cache-service   replicated          3/3                 artifacts.versa-networks.
com:8443/cache-service:latest
s8h1oiwokans        kafka_broker1             replicated          1/1                 artifacts.versa-networks.com:8443/
ecp-kafka:2.5.0             *:9092->9092/tcp
qlf6b78z2vax        kafka_broker2             replicated          1/1                 artifacts.versa-networks.com:8443/
ecp-kafka:2.5.0             *:9093->9093/tcp
8xzygy5nod59        kafka_broker3             replicated          1/1                 artifacts.versa-networks.com:8443/
ecp-kafka:2.5.0             *:9094->9094/tcp
b7a5gye8a6md        kafka_zookeeper1          replicated          1/1                 artifacts.versa-networks.
com:8443/zookeeper:3.6.2
sionbhnq2ec4        kafka_zookeeper2          replicated          1/1                 artifacts.versa-networks.
com:8443/zookeeper:3.6.2
jodrmyecmv9r        kafka_zookeeper3          replicated          1/1                 artifacts.versa-networks.
```

```
com:8443/zookeeper:3.6.2
2tzvenut4jjv        misc_mgmt-service          global            3/3               artifacts.versa-networks.com:8443/
mgmt-service:latest         *:8447->8447/tcp
sfd9wty3wmzl        misc_status-checker        global            3/3               artifacts.versa-networks.
com:8443/busybox:latest
kvcm9y2x8pwa        postgres_database-service  global            3/3               artifacts.versa-networks.
com:8443/ecp-patroni-async:2.0.1     *:5432-5433->5432-5433/tcp
dcf3i4wfmtnz        postgres_postgres1         replicated        1/1               artifacts.versa-networks.
com:8443/ecp-patroni-async:2.0.1
rpc2qanky1ce        postgres_postgres2         replicated        1/1               artifacts.versa-networks.
com:8443/ecp-patroni-async:2.0.1
9opdf3quildj        postgres_postgres3         replicated        1/1               artifacts.versa-networks.com:8443/
ecp-patroni-async:2.0.1
pv9h48jnhc8s        solr_search-service        replicated        1/1               artifacts.versa-networks.
com:8443/solr:8.4.1-slim
v2jwb48jdn1i        traefik_loadbalancer       global            3/3               artifacts.versa-networks.com:8443/
traefik:v2.3.6
```

- **docker service ps --no-trunc** *service-name*—Display information about a specific Docker service.

```
admin@concerto-1:$ docker service ps ecp_core-service
ID                  NAME                IMAGE                                                NODE                DESIRED
STATE       CURRENT STATE           ERROR               PORTS
gklzpq8bezs7        ecp_core-service.1  artifacts.versa-networks.com:8443/core-service:latest   concerto-
1           Running             Running 9 minutes ago
```

- **docker container ls –a**—List all containers running on the system.
- **docker container inspect** *container-id*—Display details about a specific container.
- **docker image ls –a**—List all Docker images loaded on the system.
- **docker volume ls**—List all Docker volumes on the system.
- **docker network ls**—List all Docker networks on the system.
- **docker events --filter 'scope=swarm'**—View Docker swarm events.
- **gluster volume status ecp-share**—Display details about the GlusterFS mounted volume. ecp-share is the name of the default volume created in Concerto cluster.

Troubleshoot Patroni

To check the status of the database in multinode deployments, issue the following command:

```
vsh database status
+ Cluster: versaecp (6963705191824814110) --+----+-----------+
| Member    | Host      | Role    | State   | TL | Lag in MB |
+-----------+-----------+---------+---------+----+-----------+
| postgres1 | 10.0.1.39 | Leader  | running | 23 |           |
| postgres2 | 10.0.1.38 | Replica | starting| 21 |       500 |
| postgres3 | 10.0.1.26 | Replica | running | 23 |         0 |
+-----------+-----------+---------+---------+----+-----------+
```

If the lag value is greater than 100 MB, or if the timeline (TL) is behind others, the replica might not be considered for

leader promotion. This might happen because of network issues between data centers. Try recovering by reinitializing the appropriate replicas. When prompted, enter the name of the member to reinitialize and recreate the replica.

```
vsh database reinit
+ Cluster: versaecp (6963705191824814110) --+----+-----------+
| Member    | Host      | Role    | State   | TL | Lag in MB |
+-----------+-----------+---------+---------+----+-----------+
| postgres1 | 10.0.1.39 | Leader  | running | 23 |           |
| postgres2 | 10.0.1.38 | Replica | starting| 21 |       500 |
| postgres3 | 10.0.1.26 | Replica | running | 23 |         0 |
+-----------+-----------+---------+---------+----+-----------+
Which member do you want to reinitialize [postgres3, postgres1, postgres2]? []: postgres2
```

This issue might occur in the following scenarios:

- Network latency to that replica may be very high. To check the latency:
  1. Issue the **labels** command to identify the node hostname. In the example output above, the **labels** command output for node3 corresponds to postgres3.
  2. Log in to the ssh console of node3/postgres3 as the admin user.
  3. From the node3 console, issue the **sudo ping –s 1475** *leader-host-ip-address*/*node1-ip-address* command to check the latency. If the latency is greater than 40 milliseconds, this is the root cause of the issue.
  4. Contact your network administrator so that they can take measures to reduce the latency.
- Missing record because of latency- or downtime-related replica synchronization. To check for a missing record:
  1. Issue the **labels** command to identify the node hostname. In the example output above, the **labels** command output for node3 corresponds to postgres3.
  2. Log in to the ssh console of node3/postgres3 as the admin user.
  3. Check the /var/log/ecp/postgresql/postgresql.log file.
  4. If you see an error in the logs such as “00xxxxx.history does not exist”, reinitialize the replica. The following is an example error message:

     ```
     050000000C40000027 has already been removed
     ERROR: 2022/12/12 23:17:03.707719 Archive '00000007.history' does not exist.
     ```

Troubleshoot Concerto Using Service Logs

The following table describes the service logs you can use to troubleshoot Concerto. All log files are stored in the /var/log/ecp directory.

Log	Description
cache-service	Hazelcast cache service logs

Log	Description
cli_audit.log	Audits all vsh command operations performed
core-service	Core service logs
deploy.log	Logs for Concerto cluster initialization
flyway.log	Database migration logs
install.log	Logs for Concerto bin installation
kafka	Kafka logs
mgmt-service	Management service logs
monitoring-service	Monitoring service logs
pgbackup.log	Logs corresponding to database backup andrestore operations
postgresql	Patroni and PostgreSQL logs
setup.log	Logs for Concerto service start and stop operations
solr	Solr logs
traefik	Traefik logs
upgrade.log	Logs for Concerto upgrade operation
web-service	Web service logs

Log	Description
zookeeper	Zookeeper logs

Use CA Signed Certificates

To use CA signed certificates in Concerto, you need to copy the CA signed certificate and key into the /var/versa/ecp/share/certs directory. Note that the key and certificate file must be named ecp.key and ecp.crt, respectively.

Configure a Kafka Authentication Connection on a Director Node

In Concerto Release10.1.x, you must configure the Concerto IP addresses as broker1/broker2/broker3 i nthe /etc/hosts file on the Director nodes. For example:

```
cat /etc/hosts
127.0.0.1       localhost
10.48.7.81      broker1
10.48.7.82      broker2
10.40.30.80     broker3


# The following lines are desirable for IPv6 capable hosts
::1     localhost ip6-localhost ip6-loopback
ff02::1 ip6-allnodes
ff02::2 ip6-allrouters
```

In Releases 10.2.x and later, you do not need to configure the Concerto IP addresses in the /etc/hosts file.

Supported Software Information

Releases 10.2.1 and later support all content described in this article.