

# OCI Setup and Configuration

## Disclaimer

This software is not officially supported by Kentik and is considered pre-alpha. Please reach out to [mikek@kentik.com](mailto:mikek@kentik.com) with any questions or requests.

## Minimum Requirements:

A valid login to cloud.oracle.com with the ability to configure logging for VCNs along with the ability to create logstreams.

Compute resources that can run an agent.

A capable system would be:

- 4 cores x86 CPU
- 8GB RAM
- 80GB disk for operating system
- Linux operating system
- Docker

Additional CPU/Memory resources may be needed for higher flow volumes.

## Create flowlogs

The process of creating flow logs in OCI is documented [here](#).

## Create logstream

Creating a logstream in OCI is outlined [here](#)

## Gather kafka information needed to connect to your OCI Stream

Once a logstream has been created, you will need to gather information to allow you to connect to it. This portion of the guide was created with the help of this [OCI quickstart guide](#).

OCI log streams are compatible with Kafka clients . This software uses the Python confluent\_kafka client, rather than using the OCI SDK, to connect to the log stream and receive log data.

As such, the only information needed for this connection are the following:

```
security.protocol: 'SASL_SSL'  
sasl.mechanism: 'PLAIN'  
bootstrap.servers  
security.protocol  
sasl.username  
sasl.password  
topic
```

## Bootstrap servers

should use your “Messages Endpoint” listed for the stream.

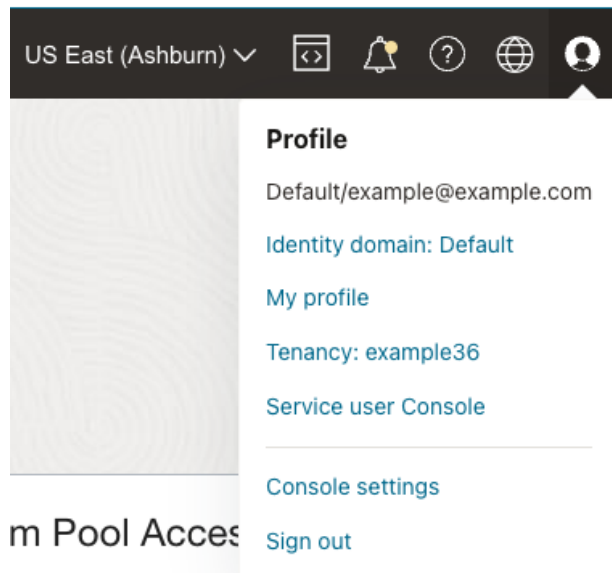
The screenshot shows the OCI console interface for a Stream Pool named 'DefaultPool'. On the left, there is a large green circle with the letters 'SP' in white, and the word 'ACTIVE' below it. To the right of the circle, the title 'DefaultPool' is displayed. Below the title are four buttons: 'Edit Settings', 'Move Resource', 'Add tags', and 'Delete'. Further right, there are two tabs: 'Stream Pool Information' (which is selected) and 'Tags'. The 'Stream Pool Information' tab contains two columns of data. The left column, titled 'Stream Pool Information', lists the Name (DefaultPool), OCID (ocid1.streampool.oc1.iad.amaaaaaa26eehyqarsgfj3ffqjcbx3udn4u6lkkxwexample123), Compartment (example36 (root)), and Encryption Key (Oracle-managed Key). The right column, titled 'Stream Pool Access', lists the Endpoint Access (Public) and the FQDN (cell-1.streaming.us-ashburn-1.oci.oraclecloud.com). Links for 'Hide', 'Copy', and 'Assign' are also present.

Stream Pool Information	
<b>Name:</b> DefaultPool	<b>Endpoint Access:</b> Public
<b>OCID:</b> ocid1.streampool.oc1.iad.amaaaaaa26eehyqarsgfj3ffqjcbx3udn4u6lkkxwexample123	<b>FQDN:</b> cell-1.streaming.us-ashburn-1.oci.oraclecloud.com
<b>Compartment:</b> example36 (root)	
<b>Encryption Key:</b> Oracle-managed Key	

## SASL username


is constructed of 3 components separated by a /. The format is shown here

<OCI\_tenancy\_name>/<your\_OCI\_username>/<stream\_pool\_OCID>



1. OCI\_tenency name is the name of the compartment you are using the example above shows "example36".
2. Your\_OCI\_username is generally the email address you use to log into the OCI console. In the example "example@example.com" is the username
3. The stream\_pool\_OCID can be found in the stream pool details In the example, it is ocid1.streampool.oc1.iad.aaaaaaaa26eehyqarsgfj3ffqjcbx3udn4u6lkksexample123

## Stream Pool Information

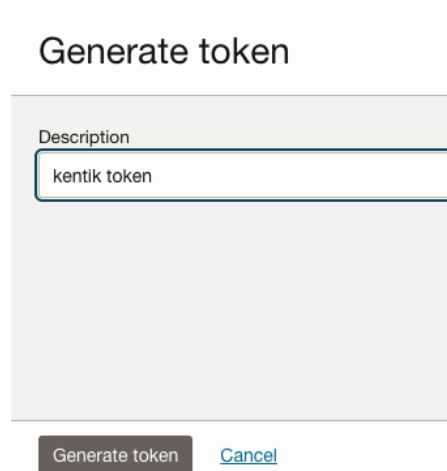
**Name:** DefaultPool 

**OCID:** ocid1.streampool.oc1.iad.aaaaaaaa26eehyqarsgfj3ffqjcbx3udn4u6lkkxwexample123 [Hide](#) [Copy](#)

In the example data given above, the complete string for the SASL username would look like this:  
example36/example@example.com/ocid1.streampool.oc1.iad.aaaaaaaa26eehyqarsgfj3ffqjcbx3udn4u6lkksexample123

## SASL password

The SASL password can be obtained by navigating to your user profile and generating an auth token. When viewing your user profile, click on "auth tokens" on the resources section of this page. Enter a name and click "generate token". This token string will be used as your sasl password in the configuration file.



Generate token

Description

kentik token

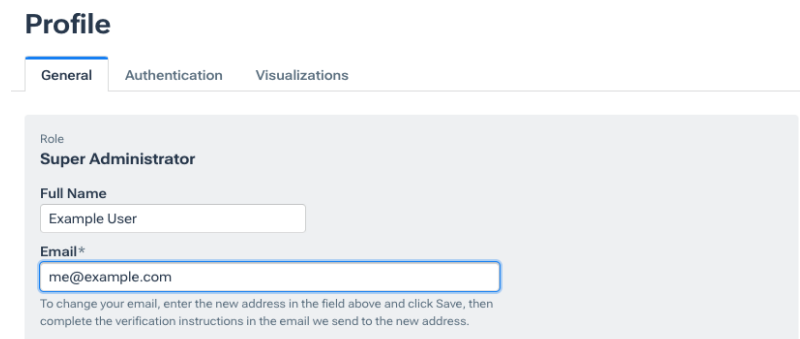
Generate token [Cancel](#)

You will use the generated token as the SASL password

## Topic

## Gather details for Kentik authentication

Go to <https://portal.kentik.com/v4/profile> In your Kentik portal. Here your authorization email will be displayed.



**Profile**

General Authentication Visualizations

Role  
**Super Administrator**

Full Name  
Example User

Email\*  
me@example.com

To change your email, enter the new address in the field above and click Save, then complete the verification instructions in the email we send to the new address.

In the authentication tab you will be able to get your API Key.

## Profile

General **Authentication** Visualizations

### Single Sign-on

SSO is enabled for your company, please use the following URL for the login page and update your bookmarks [https://portal.kentik.com/login/sso/employee\\_mike-krygeris](https://portal.kentik.com/login/sso/employee_mike-krygeris). Contact one of your Super Admins for any help with SSO.

### Password

Initiating a password reset will send you a link via email to confirm the action and setup a new password.

[Reset Password](#)

### Two-factor Authentication

[Add YubiKey](#)

[Add TOTP](#)

Enabled	Name
 No Results	



### API Token

Token

0b418fff0000ffeeefab

[Copy to Clipboard](#)

[Reset API Token](#)

## Create a device in Kentik

Before deploying your OCI device, you will need to create a device to receive flow at <https://portal.kentik.com/v4/settings/devices>

You will need to provide an IP address for the device as shown in the screen shot. This IP address just needs to be unique and distinct from any other devices. It does not have to be the IP on the device since it is just a dummy IP.

Under “Type” Select OCI Flow Log. Choose a flowpak license for this device.

## Sampling

The value entered into the device will be used by the program to sample. If you want to sample 50% of the flows, enter 2 (for 2:1 sample rate). Currently you would need to redeploy the container (or restart the service) to re-read this setting for a device.

### Flow export configuration

#### Sending IPs\*

[+ Add Sending IP](#)

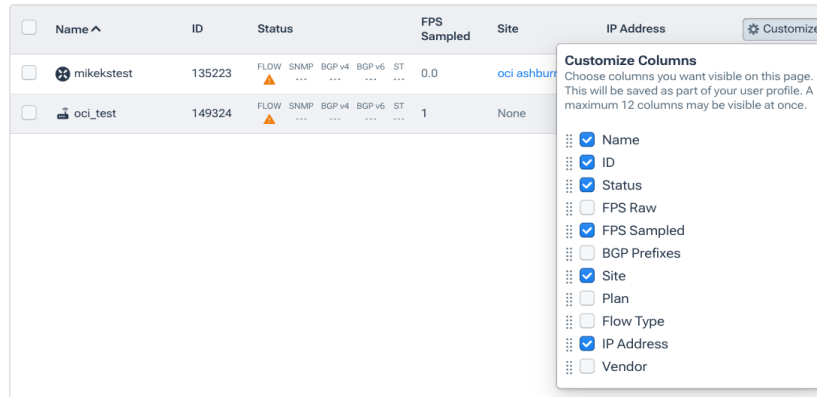
IP address(es) from which the router sends flow

#### Sample Rate\*

Total packets transiting the device for each packet processed for flow data

## Getting your device ID

In your device settings click “customize” and make sure “ID” is checked.



## Create agent configuration file

The agent configuration file must be in YAML format and named config.yaml. This will contain the credentials for each stream. Multiple config files may be created if different streams and devices are desired. All variables needed are shown <VARIABLE> format. Anything else should be left alone as a default unless instructed otherwise by Kentik.

Note: the OCI endpoint needs to keep the “:9092” port.

You may omit the sasl.password from the configuration and export an environment variable names ocitoken with the token as a value.

You may omit X-CH-Auth-API-Token from the configuration and export an environment variable names kentiktoken with the token as a value. In docker, you add -e

kentiktoken=abc12345

```
default:
  debug: 0
oci_conf:
  bootstrap.servers: "<OCI-ENDPOINT>:9092"
  security.protocol: 'SASL_SSL'
  #ssl.ca.location: '/home/user/cacert.pem'
  sasl.mechanism: 'PLAIN'
  sasl.username: '<OCI_TENANT>/<OCI_USERNAME>/OCI_STREAMPOOL_ID'
```

```
sasl.password: '<OCI-Token>'
group.id: '<UNIQUE_GROUPNAME>'
topic: '<OCI-Streamname>'
```

```
kentik_auth:
  flow_url: 'https://flow.kentik.com/chf?sender_id='
  api_url: 'https://api.kentik.com'
  X-CH-Auth-Email: '<KENTIK_USER>'
  X-CH-Auth-API-Token: '<KENTIK_API_TOKEN>'
  X-CH-App-Protocol: '0'
  Content-Type: 'application/influx'
```

```
kentik_device:
  company_id: '<KENTIK_COMPANY_ID>'
  device_name: '<KENTIK_OCI_DEVICE_NAME>'
  device_id: '<KENTIK_DEVICE_ID>'
```

## Launching as a container container

This container is hosted on [docker.io](https://hub.docker.com/r/mkrygeri/kentik-oci).

You can launch the container if the config file is in the same directory as your cli.

You can launch by running the following command:

```
docker run -d --name kentik-oci -v
$(pwd)/config.yaml:/opt/kentik/kentik-oci/config.yaml:ro
mkrygeri/kentik-oci:latest
```

Removing the container

## Running as a service

This code can also be configured to run as a linux systemd service if this is preferred.

This can be accomplished by cloning the repository and running the “service.sh” file. This file will also reinstall the the files if you perform a config change

You can pull down the source using git: `git clone mkrygeri/kentik-oci`



1. Create config.yaml as seen above(make a copy of config.example.yaml)
2. Edit the kentik.env file and put in your tokens
3. Install python3(if not installed)
4. run "pip install -r requirements.txt" in order to install any required python libraries.
5. chmod +x service.sh
6. Run ./service.sh
7. Check service by running "systemctl status kentik-oci.service"

## OCI flow log fields supported

Currently the following pieces of data are supported for ingestion into Kentik from OCI. Full definitions of these items can be found [here](#).

```
action
destinationAddress
flowid
protocolName
sourceAddress
version
datetime
id
status
compartmentid
ingestedtime
loggroupid
logid
tenantid
vniccompartmentocid
vnicocid
vnicsubnetocid
specversion
time
type
bytesOut
packets
protocol
sourcePort
startTime
destinationPort
```

## Posting data to Kentik (supplemental documentation)

This section is informational and is provided in order to inform developers on how they might implement a similar tool in a preferred language. Since this method uses HTTPS POST, it can be fairly simple for a developer to implement themselves in a different data pipeline.

This tool was created to provide an example of how to post data to Kentik using our [Influx Line Protocol](#) ingest method. Influx is just the transport method and does NOT connect to InfluxDB. Please keep this in mind. There are certain aspects to this that are required in order to do this.

1. Headers - Headers MUST contain the following information in order to properly identify the account as well as the content of the payload. Bad headers will still result in a “200” response from Kentik, so it is important that these are correct or data will be discarded.
  - a. X-CH-Auth-Email - This MUST be an email associated with an account in Kentik
  - b. X-CH-Auth-API-Token - This MUST be a valid Kentik API token associated with an account in Kentik.
  - c. Content-Type - this MUST be set to 'application/influx'
  - d. X-CH-App-Protocol - This MUST be set to '0'
2. Data - The flow data in OCI flowlogs arrives as a structured JSON object. The field definitions are defined [here](#). Influx Line Protocol dictates a rather flat listing of key value pairs.

Note: There are 4 sections of Influx records metric, metric tags, metric values and a timestamp. However, Kentik does not distinguish between metric tags and values. So you cannot force a metric tag to be a metric value, or a tag to be a value. Kentik will also disregard the timestamp at the end of the record and use the ingest timestamp, so it is generally more efficient to use a '0' for this.

## Example POST using CURL

```
curl https://flow.kentik.com/chf?sender_id={customer_id}%3A{DEVICE_NAME}%3A{DEVICE_ID}  
--data-binary @oci_flow_log_test.txt -H "X-CH-Auth-Email: {KENTIK_EMAIL}" -H  
"X-CH-Auth-API-Token: {KENTIK_TOKEN}" -H "Content-Type: application/influx" -H  
"X-CH-App-Protocol: 0"
```

The example is posting a file on disk that is in influx line format. An example of each line would look like this:

```
oci_flow_log,sampleRate=2,action=REJECT,destinationAddress=10.0.0.134  
,flowid=e8200d4c,protocolName=TCP,sourceAddress=73.218.155.16,status=  
OK,version=2,id=44422460,compartmentid=ocidl.tenancy.oc1..aaaaaaaaten  
ifiquz5kyx2h4zctz zu2gcprxohhmjx7rmywx6pz2naww6lva,ingestedtime=2023-0
```

```
3-14T02:26:58.752Z,loggroupid=ocid1.loggroup.oc1.iad.aaaaaaaa26eehyqamwasrjhpnebd7dxiwdt2glnu7rlwynsy2r5bqdqcp1a,logid=ocid1.log.oc1.iad.aaaaaaaa26eehyqa4a65snv3uo5toloty5bae6rcenrd26cdk4o5ix7zmfca,tenantid=ocid1.tenancy.oc1..aaaaaaaatenifiqz5kyx2h4zctz2u2gcprxohhmjx7rmywx6pz2naww6lva,vniccompartmentocid=ocid1.tenancy.oc1..aaaaaaaatenifiqz5kyx2h4zctz2u2gcprxohhmjx7rmywx6pz2naww6lva,vnicocid=ocid1.vnic.oc1.iad.abuwcljt7yk4kc7njt5rh2egxslt7nw7foq5a2ik5u6dale6xf65ucebi2nq,vnicsubnetocid=ocid1.subnet.oc1.iad.aaaaaaaa6cqck3wgspqjrn2o5a7wk4gun5vlbnv7i32bxasj4gj5zsbabpxq,source=-,specversion=1.0,time=2023-03-14T02:25:45.000Z,type=com.oraclecloud.vcn.flowlogs.DataEvent
bytesOut=44,destinationPort=19,endTime=1678760745,packets=1,protocol=6,sourcePort=31275,startTime=1678760745 0
```

## Other device types

There are other types of devices that can be deployed, but this is beyond the scope of this document. Please consult with Kentik's technical team for more information if you would like to use a customized data record for a different device type.

## References

Configuring flow logs

[https://docs.oracle.com/en-us/iaas/Content/Network/Concepts/vcn\\_flow\\_logs.htm](https://docs.oracle.com/en-us/iaas/Content/Network/Concepts/vcn_flow_logs.htm)

Creating log streams

[https://docs.oracle.com/en-us/iaas/Content/Streaming/Tasks/creatingstreamsandstreampools\\_create-stream.htm#create-stream-console](https://docs.oracle.com/en-us/iaas/Content/Streaming/Tasks/creatingstreamsandstreampools_create-stream.htm#create-stream-console)

Gathering OCI Stream credentials

<https://docs.oracle.com/en-us/iaas/Content/Streaming/Tasks/streaming-kafka-python-client-quickstart.htm>

Influx line protocol

<https://docs.influxdata.com/influxdb/v2.6/reference/syntax/line-protocol/>

OCI flow log format

[https://docs.oracle.com/en-us/iaas/Content/Logging/Reference/details\\_for\\_vcn\\_flow\\_logs.htm](https://docs.oracle.com/en-us/iaas/Content/Logging/Reference/details_for_vcn_flow_logs.htm)