

## Quick Docs - Deploy Observability Node CloudFormation Template

### Summary

This **quick doc** provides a quick overview of the CYLLiX Observability Node CloudFormation template, so that customers can get an idea of pre-requisites and what components are installed as part of the Template.

### Installation

The CloudFormation template creates an Observability stack as part of a previously deployed CYLLiX Apache Kafka stack.

- Prerequisite
  - Select *Choose an existing template*
- Specify template source
  - Select *Amazon S3 URL* or *Upload a template file*
- Inputs **All inputs are required to be provided despite the defaults**
  - Stack Name - Provide a meaningful name for the Stack. For e.g., *cx-observe*
  - Parameters
    - Section 1 - Network Configuration
      - KafkaVPCId- Select the CYLLiX Apache Kafka VPC from the list of VPCs displayed (Will be *kafkavpc*)
      - Route53Zone- Select the CYLLiX Apache Kafka Route53 zone. Hint: Look for the Domain name. For e.g., *cyllix.ami*
      - AZSubnet- Select one of the CYLLiX Apache Kafka Subnets.
    - Section 2 - EC2 Configuration
      - InstanceType - Specify the Instance Type – OK to select the Default – which is *t4g.large*
      - HostPrefix - Provide a host prefix for the cluster - the node# will be attached to the hostname in the creation. *Defaults to cxob*. Recommend proper naming that will easily identify the Observability host easily.

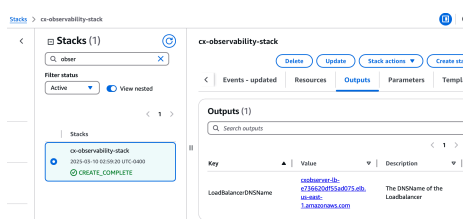
- HostDomain - Provide the *domainname* which will qualify the FQDN. For easy affiliation, select the same Domain as the Apache CYLLiX deploy - *Defaults to cyllix.ami*
  - KeyName - Select an **existing** KeyPair name from the dropdown. The KeyName should have been created earlier and downloaded (saved). The **Key** will be required for connecting to the host. **No Default Provided.**
  - Section 3 - Kafka-Cluster Egress Configuration
    - EgressCidr - Provide the CIDR range for the Egress of the Cluster. *Defaults to 0.0.0.0/0. It is strongly suggested to use this default in a public subnet*
  - Section 4 - Application Ingress Configuration
    - IngressCidr - From where Applications will connect to the Kafka-Cluster. *Defaults to 0.0.0.0/0. It is strongly suggested to not use this default in a public subnet*
- What is Created/Installed:
  - EC2
    - A single Nodes. FQDNs for the node is as per the *HostName##.DomainName* provided as input
  - EBS Volumes - Depends on the Template type (*Single* or *Dual*)
    - Single EBS volume - typically 250GB gp3
  - Network
    - Security group - created with the Ingress (*Ingress*) and Egress (*Input*) containing the Observability component ports (9090, 9093, 3000, 9100, 9115, 8080, 22)
    - Route53 private zone - created for the DNS resolution of the Node
    - Loadbalancer – A publicly accessible Loadbalancer is created. Select the URL from the *Output* from the Cloudformation Deploy
  - Systemd service for the Docker-Compose service with configuration files located in /etc/observability. This service is not enabled by default as

minimal configuration is required including enabling the service (see below in the *Post installation* section).

## Post Installation

After the install and once the instance is in a *Ready* state. Login to the instance and make the following minimal edits.

- Depending on the type of Kafka cluster that *Observability* is being setup for:
  - KRaft Install: Remove  
`/etc/observability/grafana/provisioning/dashboards/kafka_zookeeper_basic.json`
  - Zookeeper Install: Remove  
`/etc/observability/grafana/provisioning/dashboards/kafka_kraft_basic.json`
- Edit `/etc/observability/prometheus/file_sd/file_sd_services.yml` and remove or comment out entries not related to the type of Kafka Cluster.
- *Optional* edits can be made to `/etc/observability/alertmanager/alertmanager.yml` to enable Pagerduty, Slack etc. Other Registration and Setup is also required for the Pagerduty and Slack services.
  - Include the slack key in the line starting with `slack_api_url` where it says put
  - Include the Pagerduty key in the line starting with `-service-key`
- Enable the `docker-compose.service` systemd service via `systemctl enable docker-compose.service`
- Start the `docker-compose.service` system service via `systemctl start docker-compose.service`
- Verify that all components are up and running via `docker ps`
- Capture the **LoadBalancerDNSName** from the **Outputs** tab on the CloudFormation stack interface. *This will be used to access the Observability interface.*



## Support

For product support on the CYLLiX CloudFormation Observability Template and more detailed documentation, please [register](#) or [login](#) (if you have an active registration).

