EPICODE

Installation Manual

BotStream

Epicode Design Team

13th June, 2022

Version 1.2

Unit # 2201A, 22nd Floor, World Trade Centre, Rajajinagar, Bangalore - 560055
CIN: U72900KA2020PTC131771 corporate@epicode.in

1 Introduction **2**

2 Configure AWS CLI **2**

2.1 Install AWS CLI 2

2.2 Configure AWS CLI 2

2.3 Get Authentication Token 2

2.4 Create Secret 3

3 Configure NATS **4**

4 Configure IraPodWatcher/KiraPass **4**

5 Install BotStream **5**

5.1 Install BotStream Dependency Repository 5

5.2 Install BotStream 5

6 Using TestScript **6**

6.1 Requirements 6

6.2 Installation 6

7 Change History **6**

1 Introduction

The document describes the steps required to configure BotStream. All the required deb packages and yaml files mentioned in the commands can be downloaded from this link.

The installation prerequisites are:

- Kubernetes environment (preferably k3os for on premise deployments) for non telephony components.
- Server with Debian 11 for handling Telephony (BotStream)

For integration testing or for demos one can get a desktop with Debian 11 as base OS, install KVM for virtualization and create a k3os VM on it for the Kubernetes environment.

2 Configure AWS CLI

2.1 Install AWS CLI

```
curl "https://awscli.amazonaws.com/awscli-exe-linux-x86_64.zip" -o "awscliv2.zip"
unzip awscliv2.zip
sudo ./aws/install
```

2.2 Configure AWS CLI

```
$aws configure
AWS Access Key ID [None]: AKIAQKHNZ22ZHLLF6GII
AWS Secret Access Key [None]: 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
Default region name [None]: us-east-1
Default output format [None]: json
```

2.3 Get Authentication Token

The command retrieves and displays an authentication token that you can use to authenticate to an Amazon ECR public registry. Note the value generated in this step.

```
$aws ecr get-login-password --region us-east-1
```

2.4 Create Secret

Create this secret, naming it ira-ecr-user. Pass the authorization token obtained during the above step.

```
$kubectl create secret docker-registry ira-ecr-user
--docker-server=021973554866.dkr.ecr.us-east-1.amazonaws.com --docker-username=AWS
--docker-password=<use the value from the above step>
```

Note:

The authorization token is valid for 12 hours. Repeat the steps 3 and 4 to get a new token when required. You will have to delete the secret and then add it back running step 4. Run the command below to delete the secret.

```
$kubectl delete secret ira-ecr-user
```

3 Configure NATS

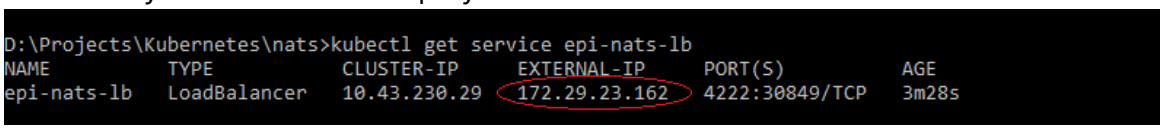
1. NATS is to be configured using NKeys which can be achieved using the following commands.

```
$helm repo add nats https://nats-io.github.io/k8s/helm/charts/
$helm repo update
$helm install epi-nats nats/nats -f nats-nkey-config.yaml
```

2. Configure NATS as a load balancer service

```
$kubectl apply -f nats-service.yaml
```

3. Once the above command is successful, get the external-ip of the loadbalancer service of your kubernetes deployment.



4. Create Config Map

   Update IRA_CLUSTER in the config map with <tenant_id> as the cluster name.

   Update IRA_NATS_WEBURL with the external-ip obtained from the above step.

   For ex:

   IRA_NATS_WEBURL: nats://172.29.23.162:4222

   Once the above changes are done in the config map, run the following command.

```
$kubectl apply -f nats-config.yaml
```

4 Configure IraPodWatcher/KiraPass

```
$kubectl apply -f watcher-depl.yaml -f tracker-depl.yaml -f kirapass-depl.yaml
```

5 Install BotStream

BotStream is released as a deb package for Debian 11 Bullseye. Procure a VM/physical machine with Debian 11 installed.

5.1 Install BotStream Dependency Repository

Download the latest version of botstream-dependency-repo deb package from this link.
Copy it to the target Debian machine and install it with the apt install command.

```
sudo apt install ./botstream-dependency-debs_1.0.0-1.deb
```

Run apt-get update after installing dependency repo so that the package manager knows about the new repo.

```
sudo apt-get update
```

5.2 Install BotStream

Download the latest version of botstream deb package from this link.
Copy it to the target Debian machine and install it using the apt install command.

```
sudo apt install ./botstream_1.1.1-1.deb
```

Enter the NATS url and cluster key when prompted for the same during the installation.

Note 1: You might see the below warning message during installation.
`W: Repository is broken: botstream:amd64 (= 1.1.1-1) has no Size information`
This can be ignored, as this does not impact the running of the software. Since apt is installing a manually downloaded package and not something from a package repo, it will not find the package's size information which will typically exist in a repo's package list.

Note 2: If you wish to change the NATS url and/or cluster key at a later date, run:
`sudo dpkg-reconfigure botstream`

6 Using TestScript

A test script is available to test out BotStream API's.

6.1 Requirements

1. Python 3.8+

6.2 Installation

1. Install python dependencies from requirements.txt using pip
2. Edit env.cmd and point NATS_SERVER_URL to the nats url.
3. Change WEBSOCKET_SERVER_HOST to the IP address of the machine.
4. Run env.cmd (windows). Change set to export if running from Linux.
5. Run server.py

7 Change History

Version 1.1 : Changes to 6.2 to add WEBSOCKET_SERVER_HOST details (15-05-2022)

Version 1.2: Changes to 1 and 5 to add prerequisites and local repo deb (13-06-2022)