Installation Manual for the Tools Used in the ZT&T System

To be installed in the Monitoring PC.

```
### Apache Spark (pyspark) Installation - START ###

STEP 1: Install and verify java version
sudo apt install openjdk-8-jdk
java -version

STEP 2: set JAVA_HOME variable
JAVA_HOME="/usr/lib/jvm/java-8-openjdk-amd64"

STEP 3: verify JAVA_HOME variable value
echo $JAVA_HOME

STEP 4: download Apache Spark latest version from offical webiste
go to https://spark.apache.org/downloads.html and download the latest verison

STEP 5: create new variable in .bashrc as follows:
sudo nano ~/.bashrc
add the following lines at the end of the file
---------------------------------
export SPARK_HOME=~/Downloads/spark-2.4.3-bin-hadoop2.7
export PATH=$PATH:$SPARK_HOME/bin
export PATH=$PATH:~/anaconda3/bin
export PYTHONPATH=$SPARK_HOME/python:$PYTHONPATH
export PYSPARK_DRIVER_PYTHON="jupyter"
export PYSPARK_DRIVER_PYTHON_OPTS="notebook"
export PYSPARK_PYTHON=python3
export PATH=$PATH:$JAVA_HOME/jre/bin
---------------------------------

STEP 6: load the .bashrc file
source ~/.bashrc

STEP 7: verify pyspark installation
spark-shell --version

### Apache Spark (pyspark) Installation - END ###
```

```
### Prometheus Installation - START ###

STEP 1: Download the setup
wget https://github.com/prometheus/prometheus/releases/download/v2.9.2/prome-
theus-2.9.2.linux -amd64.tar.gz

STEP 2: Extract the file
tar xvzf prometheus-2.9.2.linux-amd64.tar.gz

STEP 3: Navigate to download directory
cd prometheus-2.9.2.linux-amd64/

STEP 4: Run pushGateway as Background service
./prometheus &

STEP 5: verfiy running status
http://localhost:9090

### Prometheus Installation - END ###
```

```
### Grafana Installation - START ###
STEP 1: Run the following commands
sudo apt-get update
sudo apt-get upgrade
sudo reboot

STEP 2: Add gpg key to install signed packages
sudo wget -q -0 - https://packages.grafana.com/gpg.key | apt-key add -

STEP 3:
sudo apt-key list

STEP 4: update the packages and install grafana
sudo apt-get update
sudo apt-get install grafana

STEP 5: run the following command to check the Grafana installation created this
service
cat /usr/lib/systemd/system/grafana-server.service

STEP 6: run Grafana service
sudo systemctl start grafana-server

STEP 7: check the Grafana status
sudo systemctl status grafana-server
```

```
STEP 8: check Grafana GUI
http://localhost:3000

STEP 9:Enter credientials to login. Default value are admin, admin.

### Grafana Installation - END ###
```

```
### Promtail Installation - START ###

STEP 1: go to usr/local/bin
cd /usr/local/bin

STEP 2: download the pormtail
curl -O -L "https://github.com/grafana/loki/releases/download/v2.4.1/promtail-
linux-amd64.zip"

STEP 3: Extract the download file
unzip "promtail-linux-amd64.zip"

STEP 4: Allow exectable permission
sudo chmod a+x "promtail-linux-amd64"

STEP 5: create configuraiton file and add following script
sudo nano config-promtail.yml
----------------------------------
server:
  http_listen_port: 9080
  grpc_listen_port: 0

positions:
  filename: /tmp/positions.yaml

clients:
  - url: 'http://localhost:3100/loki/api/v1/push'

scrape_configs:
  - job_name: system
    static_configs:
      - targets:
          - localhost
        labels:
          job: varlogs
```

```
          __path__: /var/log/*log
----------------------------------

STEP 6: Add user for pormtail
sudo useradd --system promtail

STEP 7: create service for promtail and add the following script
sudo nano /etc/systemd/system/promtail.service
----------------------------------
[Unit]
Description=Promtail service
After=network.target

[Service]
Type=simple
User=promtail
ExecStart=/usr/local/bin/promtail-linux-amd64 -config.file /usr/local/bin/config-
promtail.yml

[Install]
WantedBy=multi-user.target
----------------------------------

STEP 8: start and check the promtail sevice status
sudo service promtail start
sudo service promtail status

STEP 9: To verify the working of promtail
curl "127.0.0.1:9080/metrics"

### Promtail Installation - END ###
```

```
### Pushgateway Installation - START ###

STEP 1: download the setup
wget
https://github.com/prometheus/pushgateway/releases/download/v0.8.0/pushgateway-
0.8.0.linux-amd64.tar.gz

STEP 2: Extract the file
tar xvzf pushgateway-0.8.0.linux-amd64.tar.gz

STEP 3: Navigate to extracted folder
```

```
cd pushgateway-0.8.0.linux-amd64/

STEP 4: Run the Pushgateway as background sevice
./pushgateway &

STEP 5: Check running status
hhtp://localhost:9091

### Pushgateway Installation - END ###
```

```
### Loki Installation - START ###

STEP 1: go to usr/local/bin
cd /usr/local/bin

STEP 2: download the loki
curl -O -L "https://github.com/grafana/loki/releases/download/v2.4.1/loki-linux-amd64.zip"

STEP 3: Extract the setup
unzip "loki-linux-amd64.zip"

STEP 4: Allow execuation permission
chmod a+x "loki-linux-amd64"

STPE 5: create onfiguraiton file for loki and paste the following content
sudo nano config-loki.yml
-------------------------
auth_enabled: false

server:
  http_listen_port: 3100
  grpc_listen_port: 9096

common:
  path_prefix: /tmp/loki
  storage:
    filesystem:
      chunks_directory: /tmp/loki/chunks
      rules_directory: /tmp/loki/rules
  replication_factor: 1
  ring:
    instance_addr: 127.0.0.1
```

```
    kvstore:
      store: inmemory

schema_config:
  configs:
    - from: 2020-10-24
      store: boltdb-shipper
      object_store: filesystem
      schema: v11
      index:
        prefix: index_
        period: 24h

ruler:
  alertmanager_url: http://localhost:9093

# By default, Loki will send anonymous, but uniquely-identifiable usage and
configuration
# analytics to Grafana Labs. These statistics are sent to
https://stats.grafana.org/
#
# Statistics help us better understand how Loki is used, and they show us
performance
# levels for most users. This helps us prioritize features and documentation.
# For more information on what's sent, look at
# https://github.com/grafana/loki/blob/main/pkg/usagestats/stats.go
# Refer to the buildReport method to see what goes into a report.
#
# If you would like to disable reporting, uncomment the following lines:
#analytics:
#  reporting_enabled: false
-------------------------

STEP 6: Add loki user
sudo useradd --system loki

STEP 7: create loki service with the following script
sudo nano /etc/systemd/system/loki.service
------------------------
[Unit]
Description=Loki service
After=network.target

[Service]
Type=simple
```

```
User=loki
ExecStart=/usr/local/bin/loki-linux-amd64 -config.file /usr/local/bin/config-
loki.yml

[Install]
WantedBy=multi-user.target
-------------------------

STEP 8: start and check the loki service
sudo service loki start
sudo service loki status

### Loki Installation - END ###
```

```
### Blockchain Agent with Prometheus - START ###

STEP 1: Activate the metrics on Blockchain
go the Besu configuration file and changed telemetry to true

STEP 2: connect with Prometheus
open the Prometheus configuraiton file and add the following lines of code under
configs
-----------------------------------------------------
- job_name: besu
  honor_timestamps: true
  scrape_interval: 5s
  scrape_timeout: 5s
  metrics_path: /metrics
  scheme: http
  follow_redirects: true
  static_configs:
  - targets:
    - node ip:port
-----------------------------------------------------

STEP 3: restart the prometheus service

STEP 4: verify the connectivity of blockchain agent with Prometheus
go to localhost:9090 --> click Status --> click Targets
It should show the blockchain/merics as scrape target

### Blockchain Agent with Prometheus - END ###
```

```
### Ziti Agent with Prometheus - START ###

STEP 1: Check expose metrics by ziti
curl <ziti controller ip>:1280/metrics

STEP 2: Open prometheus yaml file
sudo nano <prometheusrootfolder>/prometheus.yml

STEP 3: configure ziti end-point with Prometheus
Add the following lines inside the static_configs
----------------------------------------
- job_name: ziti
  honor_labels: true
  honor_timestamps: true
  scrape_interval: 5s
  scrape_timeout: 5s
  metrics_path: /metrics
  scheme: https
  tls_config:
    insecure_skip_verify: true
  follow_redirects: true
  static_configs:
  - targets:
    - 172.18.102.169:1280
----------------------------------------

STEP 4: verify the connectivity of ziti agent with Prometheus
go to localhost:9090 --> click Status --> click Targets
It should show the ziti/merics as scrape target

### Ziti Agent with Prometheus - END ###
```