BrowsableStoreAnalytics

This is a project that demonstrates how virtual reality may be used to gather better analytics from users when evaluating environments. Many companies maintain large spaces in which to prototype store layouts and designs. Constructing spaces in virtual reality is cheaper and allows a company to gather more data from testing than a simple survey.

First Time Setup

Setting Up Prerequisites

This section taken from https://calben.gitbooks.io/unreal-engine-as-a-research-framework/content/chapter1.html

Unreal Engine

This may come as some surprise, but start by installing Unreal Engine (UE4). There are good instructions on how to do this here: https://docs.unrealengine.com/latest/INT/GettingStarted/Installation/

Visual Studio 2017

As of date of publication, Visual Studio (VS) 2017 is the latest version of VS, so you can pick this up here: https://www.visualstudio.com/downloads/. When you do the install, make sure to go to the "Games" tab and install the C++ features available there (TODO go back and find the exact set of features that need to be selected to ensure everything will work during the monthly reset)

Docker Toolbox for Windows

We will use the Docker Toolbox to host our TICK stack. You may pick up the Docker Tools here: https://www.docker.com/products/docker-toolbox. When installing, choose to install Kitematic. If you have HyperV enabled on your machine, you can use the HyperV implementation. If you don't have HyperV running (or have the Home version of Windows), then you're going to need Virtualbox.

Setting Up the TICK Stack

This section taken from https://calben.gitbooks.io/unreal-engine-as-a-research-framework/content/setting-up-the-tick-stack.html

We're going to set up the TICK stack using Docker. If you wanted to install it on a remote server, you could follow the documentation here:
https://docs.influxdata.com/chronograf/v1.3/introduction/getting-started/

What Is TICK?

TICK stands for Telegraf, InfluxDb, Chronograf, and Kapacitor. For convenience, I am going to quote descriptions of the TICK stack here:

Telegraf

> Telegraf is a plugin-driven server agent for collecting and reporting metrics. Telegraf has plugins or integrations to source a variety of metrics directly from the system it's running on, pull metrics from third party APIs, or even listen for metrics via a StatsD and Kafka consumer services. It also has output plugins to send metrics to a variety of other datastores, services, and message queues, including InfluxDB, Graphite, OpenTSDB, Datadog, Librato, Kafka, MQTT, NSQ, and many others.

InfluxDB

> InfluxDB is a Time Series Database built from the ground up to handle high write & query loads. InfluxDB is a custom high performance datastore written specifically for timestamped data, including DevOps monitoring, application metrics, IoT sensor data, & real-time analytics. Conserve space on your machine by configuring InfluxDB to keep data for a defined length of time, automatically expiring & deleting any unwanted data from the system. InfluxDB also offers a SQL-like query language for interacting with data.

Chronograf

> Chronograf is the administrative user interface and visualization engine of the platform. It makes the monitoring and alerting for your infrastructure easy to setup and maintain. It is simple to use and includes templates and libraries to allow you to rapidly build dashboards with real-time visualizations of your data and easily create alerting and automation rules.

Kapacitor

> Kapacitor is a native data processing engine. It can process both stream and batch data from InfluxDB. Kapacitor lets you plug in your own custom logic or user-defined functions to process alerts with dynamic thresholds, match metrics for patterns, compute statistical anomalies, and perform specific actions based on these alerts like dynamic load rebalancing. Kapacitor integrates with HipChat, OpsGenie, Alerta, Sensu, PagerDuty, Slack, and more.

Running the Setup

Assuming we're going the Docker route, you need to do the following:

1. Start up Kitematic. You will be asked to use virtualbox if the program can't find a "native" setup (HyperV). Use VirtualBox if no native setup is fine, otherwise you shouldn't see a message.
2. You will be presented with a message that says "Starting a Docker VM." This is creating a virtual machine for you called default that will be used to host your Docker containers.
3. After this runs, you're ready to set up your Docker containers! You will be asked to log in to Docker Hub. I recommend you make an account or log in, but if you'd rather not there's a "Skip For Now" hidden in the bottom right corner.
4. You should now be at the home menu for Kitematic. We're going to use the Docker command line to enter the below instructions:

```
docker network create influxdb
docker run -d -p 8086:8086 --name=influxdb --net=influxdb library/influxdb
docker run -d -p 9092:9092 --name=kapacitor -v kapacitor:/var/lib/kapacitor --net=influxdb -e KAPACITOR_INFLUXDB_0_URLS_0=http://influxdb:8086
library/kapacitor
docker run -d -p 8888:8888 --name=chronograf --net=influxdb library/chronograf --influxdb-url=http://influxdb:8086
```

If you want to know what we just did, here is a version of the commands to run with comments:

```
# create a new network called influxdb.
# docker containers connected to the image can access it with http://influxdb:<portname>
docker network create influxdb
```

```
# start an inflxudb container connected to the influxdb network with 8086 mapped (influxdb's port)
docker run -d -p 8086:8086 --name=influxdb --net=influxdb library/influxdb

# start a kapacitor container with 9092 mapped (kapacitor's port)
# and create a docker volume we can use to make tickscripts
# we also connect this to the influxdb network and force an influxdb url
docker run -d -p 9092:9092 --name=kapacitor -v kapacitor:/var/lib/kapacitor --net=influxdb -e KAPACITOR_INFLUXDB_0_URLS_0=http://influxdb:8086
library/kapacitor

# start a chronograf container with 8888 mapped (chronograf's port)
# and force an influxdb url on the influxdb network
docker run -d -p 8888:8888 --name=chronograf --net=influxdb library/chronograf --influxdb-url=http://influxdb:8086
```

Setting Up Grafana (Optional)

Grafana is an alternative to Chronograf. If you want features like pie charts, you want to use Grafana.

```
docker run -d -p 3000:3000 --name=grafana --net=influxdb grafana/grafana
```

Exposing Your Dashboard To Your Network (optional)

After finishing the above, you may install nginx to expose your docker containers to the rest of the world.

1. Download nginx for Windows http://nginx.org/en/download.html
2. Extract nginx to a directory on your machine (I like to put these sorts of things in C:\Utils\)
3. Open the conf directory in your nginx installation and edit nginx.conf
4. This is where all of your nginx server settings are
5. Go into the server section of the configuration file
6. Change listen 80; to listen *:80 (should do the same thing, but I'm paranoid on Windows)
7. In location / {} delete everything already present (should be root html index stuff)
8. In location put proxy_pass http://<your docker's ip>:<either 8888 for Chronograf or 3000 for Grafana>

Your result should look something like this:

```
........

server {
listen *:80;
server_name localhost;

#charset koi8-r;

#access_log logs/host.access.log main;

location / {
proxy_pass http://192.168.99.100:3000;
}

........
```

Docker likes to use 192.168.99.100, so there's a high likelihood but not a guarantee that it'll be the same address for you.

Making The Database

Return to the Docker CLI from Kitematic and run the below:

```
docker exec -u 0 -it influxdb bash
```

Then run influx to enter the InfluxDB shell.

```
InfluxDB shell version: 1.3.5
> CREATE DATABASE analytics
> SHOW DATABASES
name: databases
name
----
_internal
analytics
```

Setting up BrowsableStoreAnalytics

Once your virtual machines and databases are ready.

Running An Existing Installation

The BrowsableStoreAnalytics requires all of the functionality above to be actively running. This includes:

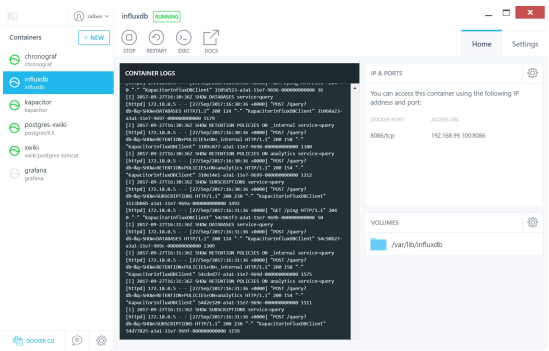
1. The InfluxDB database container on the Win64 machine

2. The Grafana server container on the Win64 machine
3. The nginx server on the Win64 machine
4. The BrowsableStoreAnalytics Unreal Engine application on the Win64 machine
5. A browser open to the Win64 machine's IP address on another device on the same network

Booting the InfluxDB Database

If the InfluxDB database isn't already running, run the Kitematic program from the start menu. After Kitematic boots, you will be presented with a screen that has a panel docked on the left side. On this panel are a few options. To make sure InfluxDB is running, select the influxdb item from the left panel and then select Start.

You should get a screen that looks similar to the following:



Booting the Grafana Server

A similar procedure to the InfluxDB Database startup should be followed. On the left hand side of the same screen should be an item named grafana. Select this item and select run.

Booting the nginx Server

A copy of nginx can be found in the BrowsableStoreAnalytics/nginx/ directory.