What the SAAS?

Jordan Kassof and Jeff Weltman

4/8/2018

Introduction

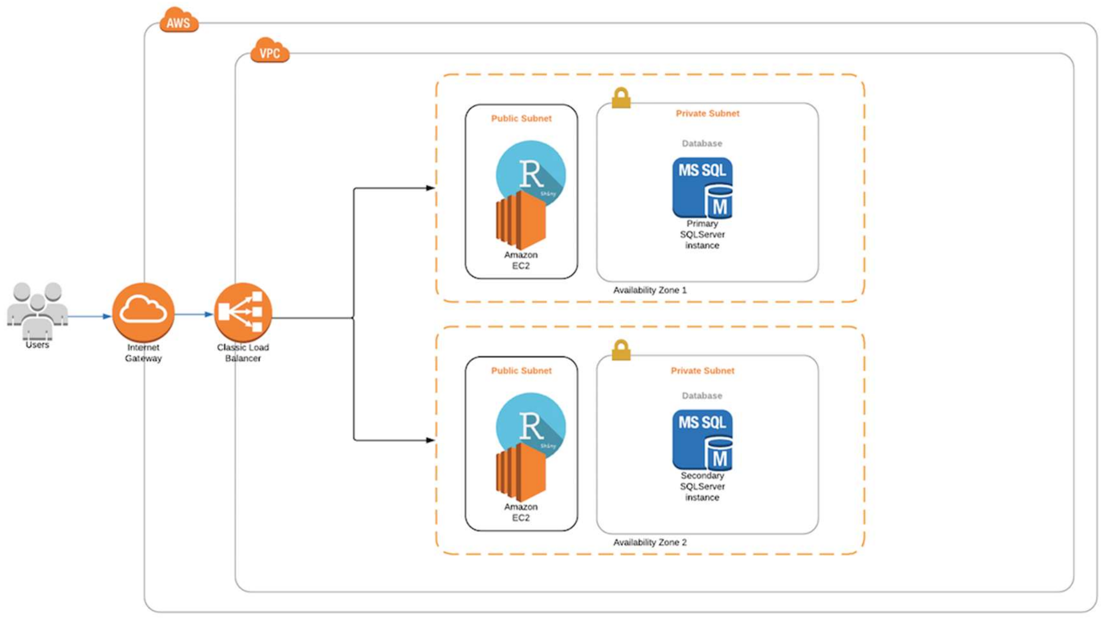
Software as a service (SAAS) is a business model that has become very familiar to most people. Many of the interactions we have with some of the most popular corporations is through a SAAS business model. Whether its for entertainment (Netflix/Hulu/AmazonPrime), socialization (Facebook/Twitter/Snapchat), financial management (Mint/Betterment), more and more firms are looking to deliver their products/services through a SAAS model.

All that being said, what is a SAAS? It's such a buzzword today, but do most regular folks really get what that means? To better understand the nature of this increasingly common arrangement, we seek to implement a small application that leverages some of the key technological components of a SAAS.

Implementation Plans

Infrastrucutre

While there are many cloud service providers, Amazon has by far the most services available on their platform. We will take advantage of just a few of those services in this implementation. The application itself will be hosted leveraging the Elastic Compute (EC2) service and the data the application uses to run will be held in a MySQL database on the Relational Database Service (RDS). In order to accomodate a large user load, we will use a load balancer to distribute incoming traffic across multiple instances of the described EC2/RDS infrastructure. Below is a diagram displaying the envisioned configuration.



Data

The Global Terrorism Database - hereafter referred to as GTD - is an open-source database of domestic, transnational, and international terrorist incidents. This database contains credibly sourced information regarding over 170,000 terrorist incidents from 1970 through 2016. For recent incidents, this data includes 135 quantitative and categorical variables describing all manner of information regarding the incident itself. These points include information regarding the weapons used, number of fatalities, amount of property damage involved, and whether any organization or individual(s) claimed credit for the incident. The majority of these cases are bombings, assassinations, and kidnappings. The GTD is maintained by the University of Maryland's *National Consortium for the Study of Terrorism and Responses to Terrorism (START).* START makes the data available both to government representatives and any interested parties.

We opted to filter the data to only consider those incidents which were both confirmed acts of terrorism and were successfully executed. Furthermore, for data visualization, we excluded those events which did not have latitude and longitude coordinates. This brought the number of incidents examined down from over 170,000 to roughly 113,000. Thus the app detailed below demonstrates visualization methods for displaying confirmed and successful terrorist incidents for which precise geographical information was available.

Application

Shiny is an R package for building interactive web apps from within RStudio. These apps can be embedded in R Markdown documents, hosted as standalone apps on a webpage, or built as dashboards. In addition, they can be extended with JavaScript, CSS themes, and htmlwidgets. Many R packages are developed both for local use and integration with Shiny for interactivity. As the GTD data subset we used contained latitude and longitude coordinates, a mapping package was a logical and appropriate choice.

We chose **Leaflet**, an R wrapper for the widely-used JavaScript library of the same name. Leaflet allows for highly-customizable map widgets with parameters for the widget itself and the user interface. Some of the many functions of Leaflet include chloropleths, GeoJSON and TopJSON maps, multi-layered maps, projections, and - with the addition of the **leaflet.extras** package - heatmaps. Many of these functions support markers and circle markers as ways to provide both interactivity and a richer source of detailed information to the end user. Additionally, through the use of database connecting R packages, such as DBI, Leaflet can visualize data hosted externally, such as in the present example of an AWS-hosted RDS instance.

For the present project, we opted to include three separate Leaflet visualizations in our Shiny web app. Upon launching the app, the end user is presented with a world map devoid of any markers or data. The *Please select a dataset* dropdown menu presents three choices:

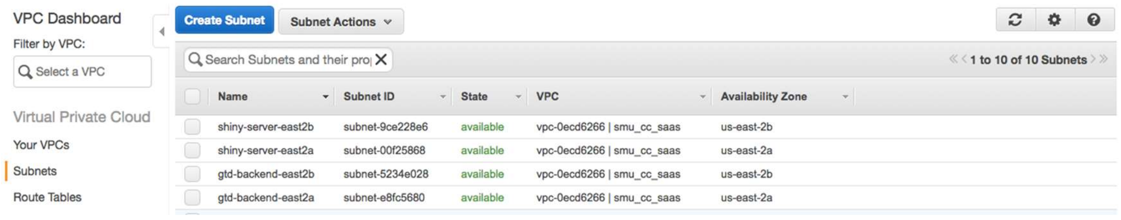
1. *Total Successful Incidents* - a heatmap of all 113,000 successful and confirmed terrorist incidents, grouped by latitude and longitude coordinates.
2. *Total Incidents By Decade* - a heatmap of successful and confirmed terrorist incidents, grouped by both decade and coordinates.
    a. Selecting this dataset shows an otherwise-hidden secondary dropdown menu for decade selection.
    b. By default, the heatmap will show incidents form the 1970s.
3. *Incidents With Excessive Property Damage* - a map of successful and confirmed terrorist incidents which resulted in at least $1,000,000 of property damage.
    a. These incidents are identified via clustered map markers. Upon mouseover, the geographical bounds of the clustered region is indicated.Clicking on an individual marker displays a popup which provides information regarding the incident, perpetrator, and estimated damage.

These three visualizations demonstrate some of the capabilities offered through Shiny and Leaflet avaiable in a SAAS offering. Locally and remotely-hosted data can be transformed in R and then elegantly displayed to provide interactive and actionable visualization utilizing Shiny and Leaflet.

Results

AWS Application Infrastructure

It was our experience that when building out AWS infrastructure, it is helpful to a) have a complete plan from the start and b) focus on building from the outside in – setting up “containers”, then filling those containers with the correct pieces. For this project, we started by making our VPC, the large “container” that everything will be in. Next, we make four subnets, two that will house app servers and two that will house backend databases.

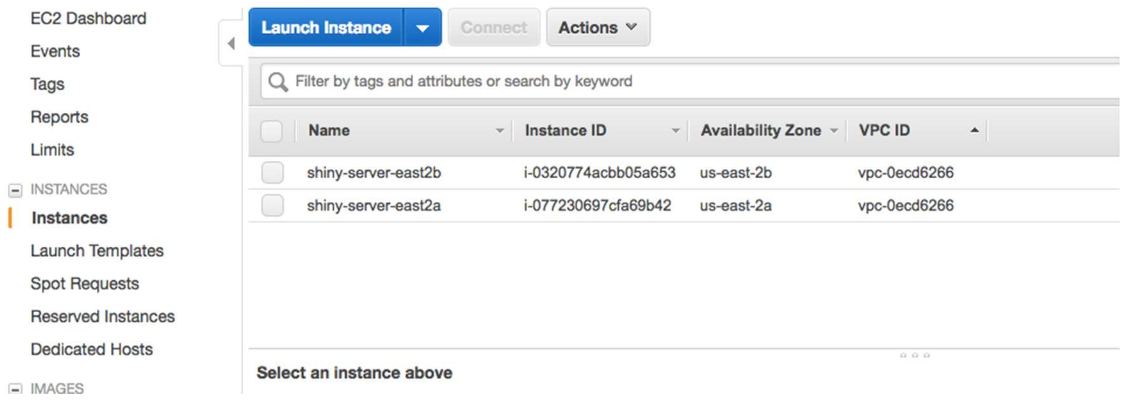


Name	Subnet ID	State	VPC	Availability Zone
shiny-server-east2b	subnet-9ce228e6	available	vpc-0ecd6266 | smu_cc_saas	us-east-2b
shiny-server-east2a	subnet-00f25868	available	vpc-0ecd6266 | smu_cc_saas	us-east-2a
gtd-backend-east2b	subnet-5234e028	available	vpc-0ecd6266 | smu_cc_saas	us-east-2b
gtd-backend-east2a	subnet-e8fc5680	available	vpc-0ecd6266 | smu_cc_saas	us-east-2a

AWS Subnets

We place each pair of subnets in different availability zones so that if an AWS availability zone is having problems, our application can still run off the other zone.

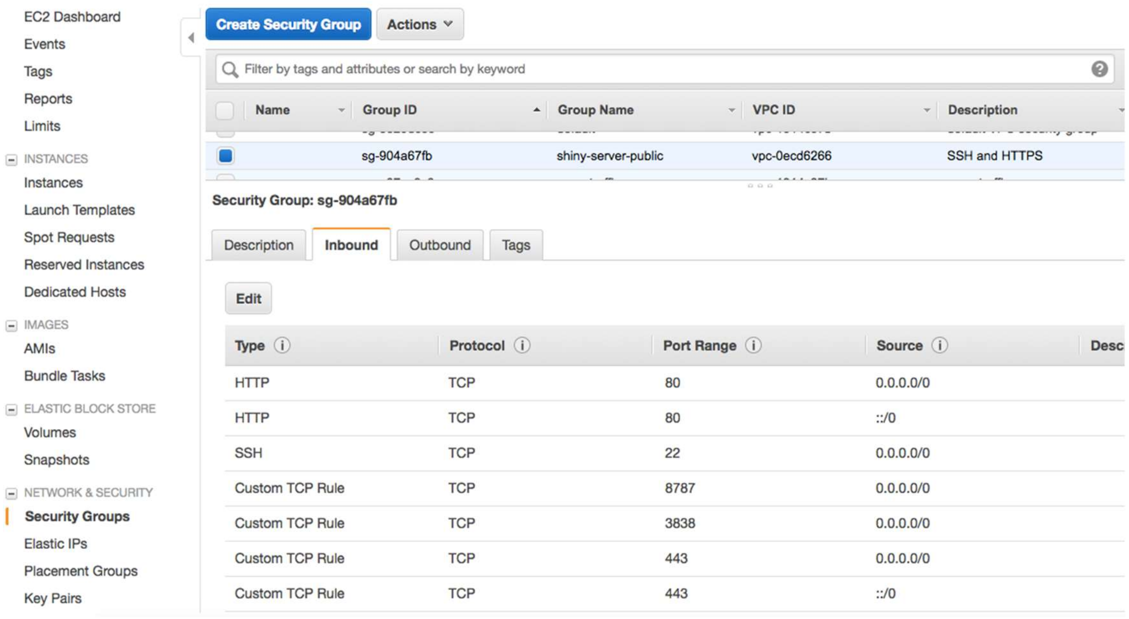
Next, we set up our two app servers, placing them into the subnet 'containers' we just made.



Name	Instance ID	Availability Zone	VPC ID
shiny-server-east2b	i-0320774acbb05a653	us-east-2b	vpc-0ecd6266
shiny-server-east2a	i-077230697cfa69b42	us-east-2a	vpc-0ecd6266

EC2 App Servers

Since we want the application to be available through normal web browsers, we need to set up a security group that allows http/https access to the servers on ports 80 and 443. Shiny hosts applications on 8787 and 3838 so those are open as well. We are also temporarily allowing SSH access into the server which will allow us to configure the servers and deploy the application.



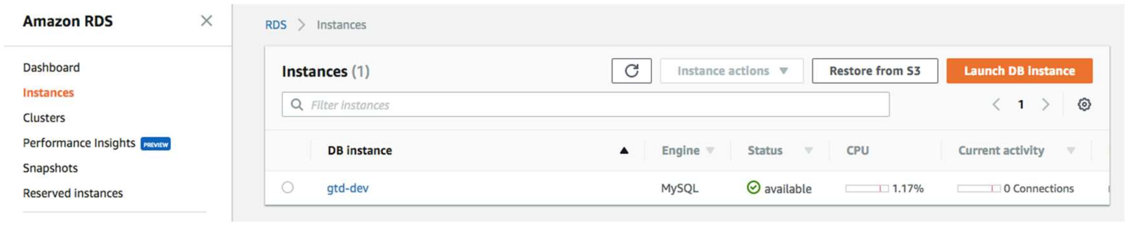
Name	Group ID	Group Name	VPC ID	Description
	sg-904a67fb	shiny-server-public	vpc-0ecd6266	SSH and HTTPS

Security Group: sg-904a67fb

Type	Protocol	Port Range	Source	Desc
HTTP	TCP	80	0.0.0.0/0	
HTTP	TCP	80	::/0	
SSH	TCP	22	0.0.0.0/0	
Custom TCP Rule	TCP	8787	0.0.0.0/0	
Custom TCP Rule	TCP	3838	0.0.0.0/0	
Custom TCP Rule	TCP	443	0.0.0.0/0	
Custom TCP Rule	TCP	443	::/0	

Security Groups for App Servers

Finally, we set up the RDS database backend using a MySQL instance. This database and one like it go into the remaining two subnets.



DB instance	Engine	Status	CPU	Current activity
gtd-dev	MySQL	available	1.17%	0 Connections

Database Backend

And that's it! We now have a functional (albeit empty) cloud infrastructure that facilitates hosting an application with a database backend. Next step will be setting up the app servers and deploying the application.

Data & Application

Further development

Due to the rich development history and active user community, any R-based application has an innumerable number of possibilities when it comes to future development. We have included the following three ways which are particularly relevant for this particular application.

First, with the capabilities of surfacing and visualizing potentially sensitive data, one opportunity for further development would be to implement authentication. RStudio's Shiny Pro Server includes out-of-the-box support for Goggle authentication, PAM (Pluggable Authentication Module), Kerberos, and LDAP. One or more of these options should integrate with any client's existing authentication infrastructure and would allow for clients to then authenticate in order to access data which is defined as permitted and appropriate. Secondly, since R supports a wide variety of endpoint connectors, this provides robust extinsibility in that data from other client systems can be ingested and accessed for highly customizable reports and analytics. This includes additional database environments, flat files, and third-party services. Lastly, client systems could be configured to automatically populate available data points in a templatized fashion. This could provide a way to facilitate self-service delivery of reporting and visualization.