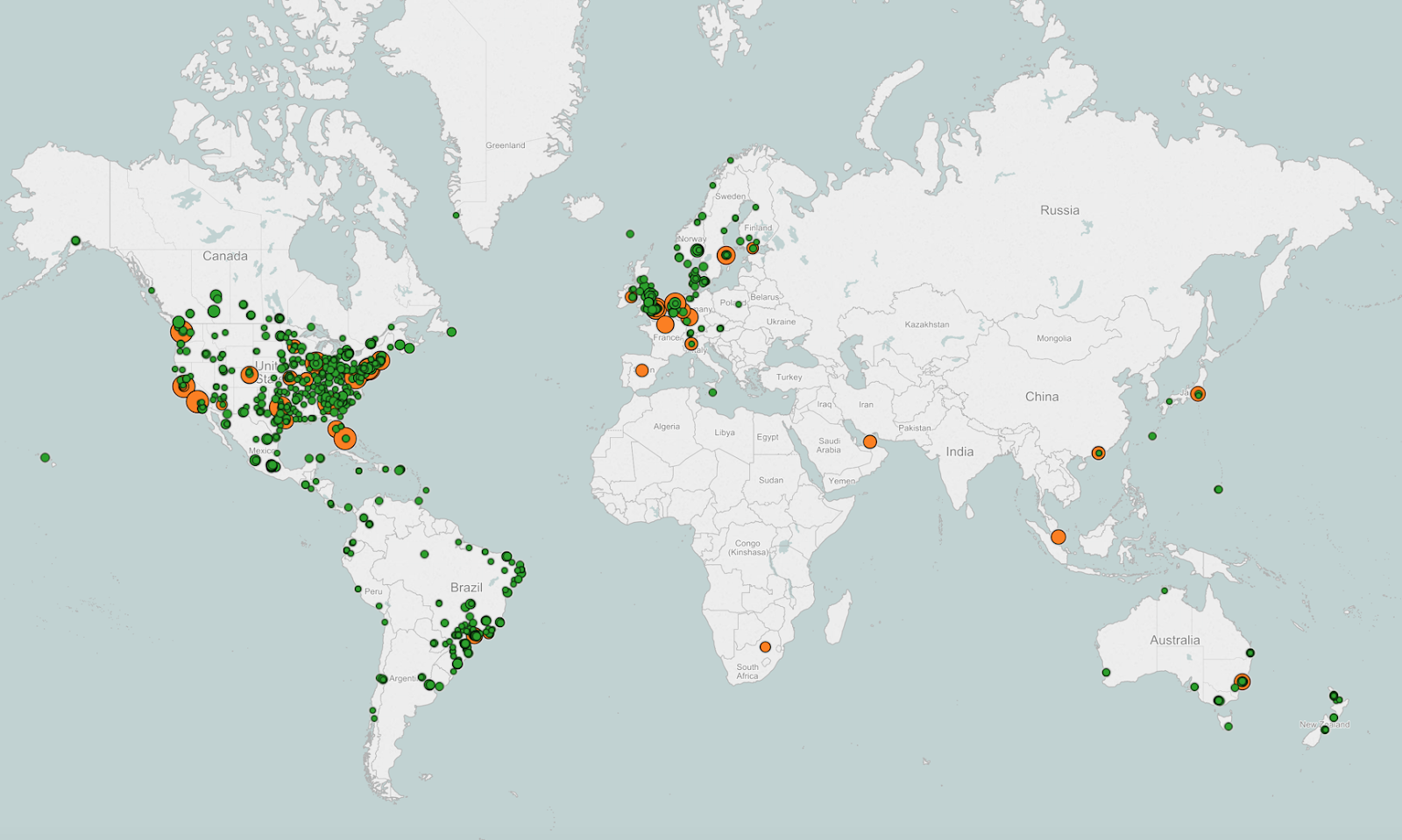
Per tal de que el nostre servei sigui el més ràpid possible i el màxim de tolerant a fallades possibles, hem optat per una solució basada en el cloud. En el cloud tenim 3 grans proveidors: Amazon Web Services, Microsoft Azure i Google Cloud Provider.

La nostre solució es basa principalment en Amazon Web Services, tot i que estem oberts a utilitzar algun altre servei d’algun altre proveidor si es creu necessari.

Arquitectura frontal web

Per aprofitar el màxim el cloud i que el temps de resposta dels nostres servidors sigui el minim possible utilitzarem instancies de EC2. Aquestes instancies de EC2, estaran repartides per tot el món i derrera Load Balancers. Aquest Load Balancers tambe estaran repartits per tot el mon, pero no tindrem tantes instancies de Load Balancers com instancies de EC2. Els Load Balancers, seguiran la seguent politica: **respondre a l’usuari en el minim temps possible**.

Per aconseguir respondre l’usuari en el minim temps possible, els load balancers reenviaran a l’usuari a la instancia/instancies de EC2 que tinguin mes aprop i que estiguin menys ocupades.



Com podem veure en la imatge, aixi es com queda la nostre arquitectura repartida per tot el mon.

Els punts taronges son les instancies de Load Balancers i els punts verds son instancies de EC2.

Com podem observar el lloc on hi ha mes instancies de EC2 es a Estats Units, aixo es degut que el 80% de les nostres son peticions son originaries d’alla.

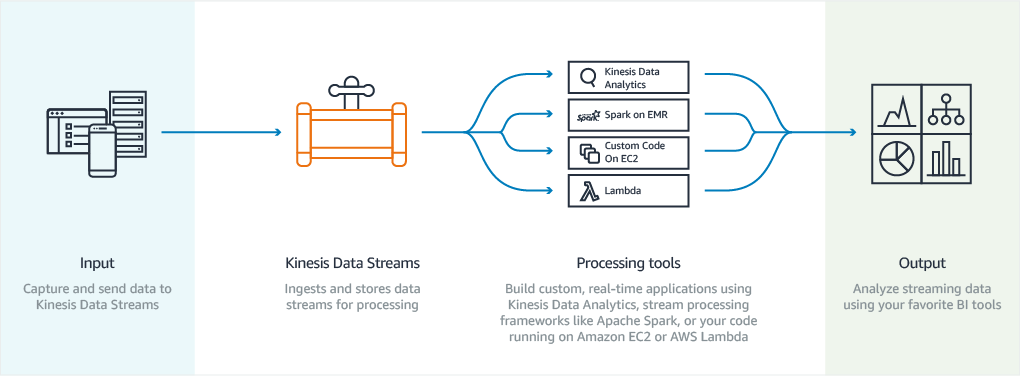
Arquitectura Backend web/mobile

Per poder servir la quantitat de instancies que tenim pel frontal web, a l’hora de fer el backend hem optat per una arquitectura serverless (sota Lambda Functions).

Aquestes Lambda functions, es Servers as a Service (SAAS). Hem optat aquesta solucio per els anomalies de clients que hi poden haver, s’escala automaticament.

Arquitectura Machine Learning

Tant l’aplicacio web, com l’aplicacio mobil continuament estaran enviant informacio a un stream de kinesis (tambe un servei de Amazon Web Services). Aquest stream de Kinesis es el encarregat de fer el machine learning sobre l’usuari, intentar aconseguir el producte que el usuari vol i que nosaltres tinguem el maxim benefici possible.



Per la part del Machine Learning, utilitzarem el servei d’Amazon Web Services anomenat Elastic Map Reduce. Aqui es on tindrem tots els nostres algorismes de Machine Learning. Aquests algorismes son els encarregats de aportar la millor Customer Experience possible. Ens ajudaran des de treure el millor producte que l’usuari que esta buscant fins a obtenir els millor sugeriments possibles per l’usuari basats en compres antigues d’aquest usuari.

Arquitectura Carrito

Per guardar els productes que te al carro, utilitzarem un Redis (a Amazon Web Services conegut com ElastiCache), on la key sera la sessio del usuari i el value els productes que te.

Utilitzarem una memcached perque volem que el temps de servir el carrito sigui el menor possible.

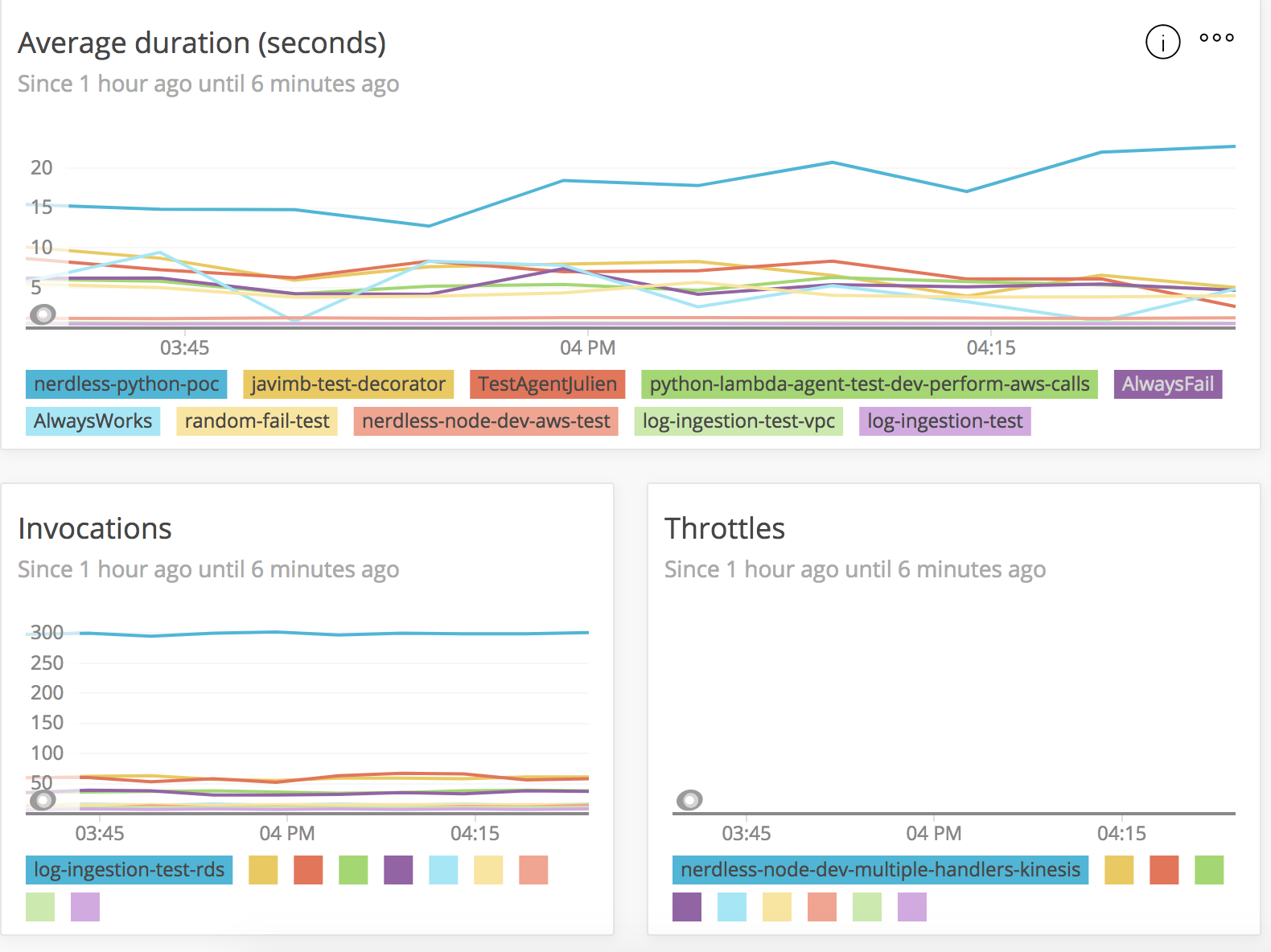
Instrumentacio

Com que la nostre solucio esta basada al 100% en el cloud, se’ns presenta el seguent problema: Com podem garantir que tots els serveis estan funcionant correctament i de la manera que nosaltres volem?

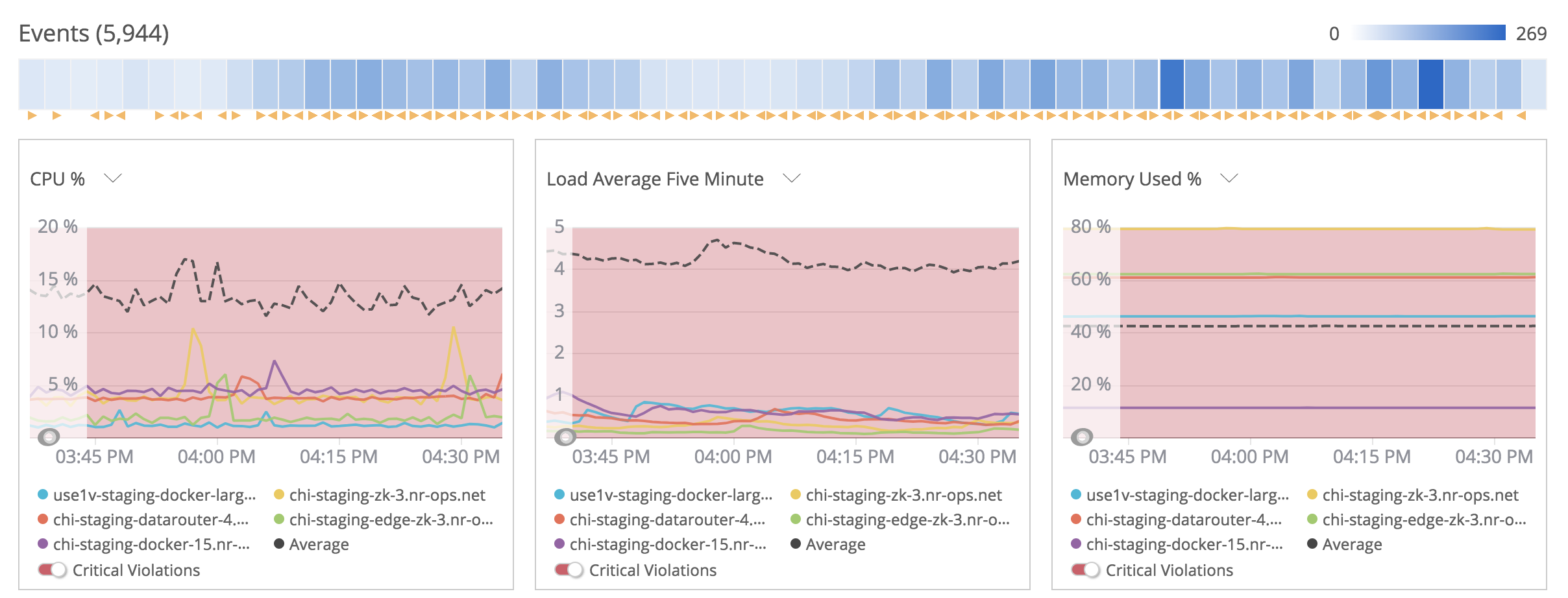
Doncs be, per monitorizar el nostre cloud farem servir una eina anomenada New Relic.

New Relic, es una eina que ens ajudara a veure si el nostre sistema esta funcionant correctament.

Per monitoritzar el funcionament utilitzarem el seu servei anomenat Cloud Integrations, aixi ens podrem programar alertes si el funcionament dels serveis no es el correcte.



Per monitoritzar les instancies de EC2, instalarem el seu agent on-host. Aixi aconseguirem treure metriques sobre les maquines, com ara: consum de network, memoria, CPU, etc.



Per monitoritzar els nostres serveis backend, utilitzarem el agent APM. Aixi podrem treure metriques sobre el temps de resposta dels servidors i veure si hi ha hagut cap error no destitgat.

