

Bitcoin Analytics

Les Enjeux

- Le Bitcoin
 - Crypto monnaie faisant de plus en plus d'adeptes
 - Monnaie très volatile avec un taux de change qui fluctue beaucoup
 - On dénombre en moyenne entre 7 et 10 transactions par seconde
- But
 - Suivre l'évolution de bitcoin en index, nombre et volume de transactions
 - Connaître les mineurs qui gagnent le plus par jour et par mois
 - Traiter un nombre important de transactions en temps réel
 - Concevoir une architecture évolutive capable de passer à l'échelle

Choix des composants logiciels

- Apache Kafka
 - Publication et consommation de données et temps réel
 - Fonctionne en mode publish/subscribe
 - Partitionnement de données permettant leur consommation en parallèle
 - Réplication de données et tolérance aux pannes
- Storm
 - Traitement des données en parallèle via l'utilisation de DAG

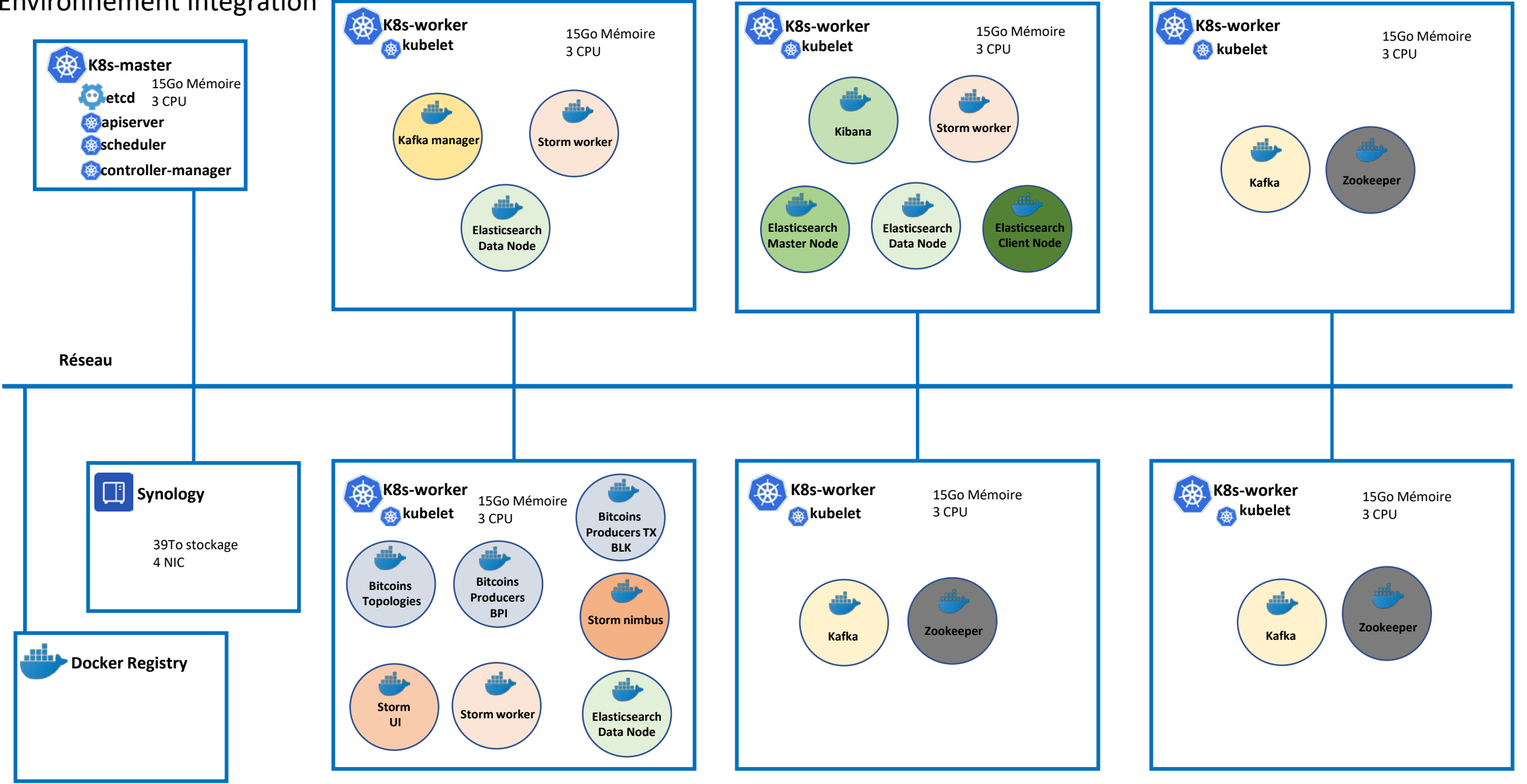
Choix des composants logiciels (suite)

- Elasticsearch
 - Base de donnée NoSQL offrant la recherche fulltext
 - Optimisé pour les calculs basés sur des agrégations
 - Partitionnement et Réplication de données, tolérance aux pannes
- Kibana
 - Outil de présentation de données
 - Optimisé pour Elasticsearch car faisant partie de la stack ELK (Elasticsearch, Logstash, Kibana)

Choix de l'architecture de la solution

- Le choix s'est porté sur Kubernetes (k8s) et Docker
 - K8s est un gestionnaire de cluster de conteneurs Linux. Il permet de gérer l'élasticité et la résilience.
 - Docker est un gestionnaire de conteneurs. Il permet l'isolation et le packaging des composants.
 - Scripts de configuration du cluster k8s et des composants techniques de la solution :
 - Création de VM : <https://github.com/plawson/kubernetes-cluster/tree/master/virtual-machines>
 - Zookeeper : <https://github.com/plawson/kubernetes-cluster/tree/master/kubernetes-zookeeper>
 - Kafka : <https://github.com/plawson/kubernetes-cluster/tree/master/kubernetes-zookeeper>
 - Storm : <https://github.com/plawson/kubernetes-cluster/tree/master/kubernetes-storm>
 - Elasticsearch, kibana : <https://github.com/plawson/kubernetes-cluster/tree/master/kubernetes-esstack>
 - NFS provisioner : <https://github.com/plawson/kubernetes-cluster/tree/master/nfs-provisioner/manifests>

Environnement Intégration



Solution

- Producteurs

- Un script python pour les transactions et les blocks.
- Un script python pour l'index du prix.
- Les scripts injectent les données récoltés dans 3 topics Kafka.
- Scripts : <https://github.com/plawson/oc-projet3/tree/master/bitcoin-producers>

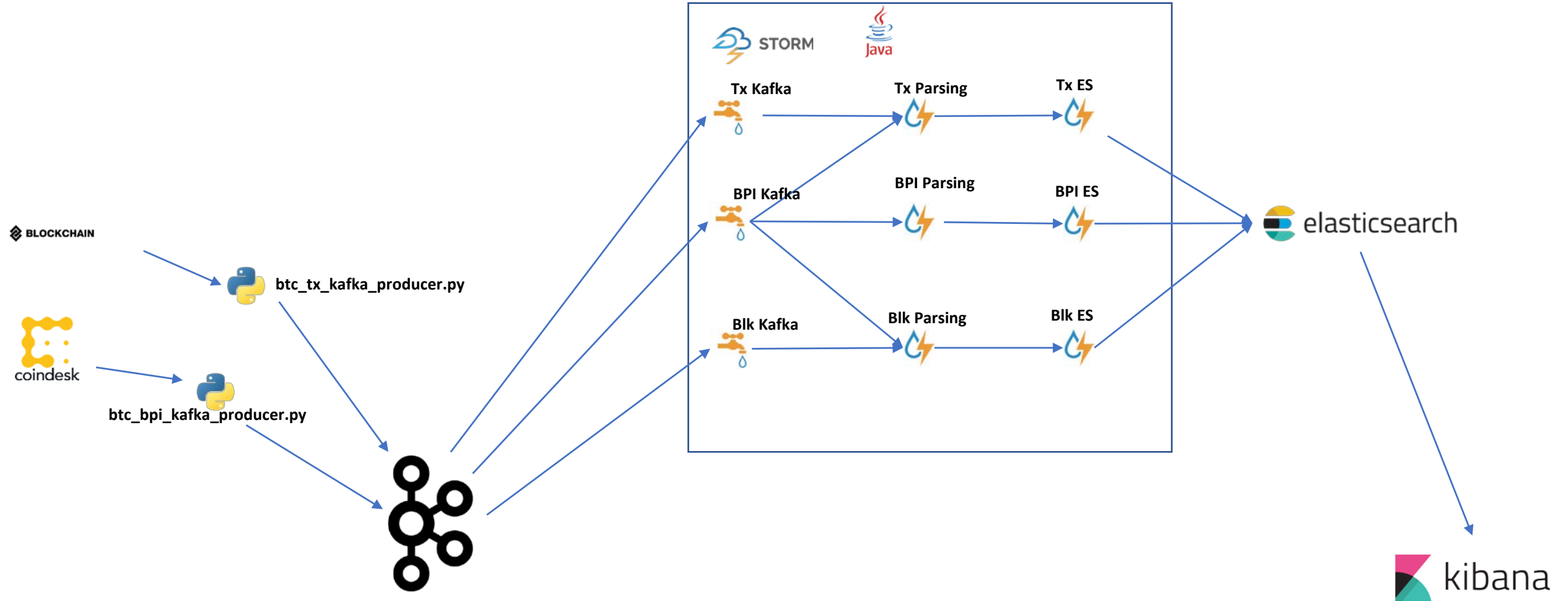
- Consommateurs

- Un programme Java crée une topologie Storm qui lit les données dans Kafka, les transforme et les injecte dans Elasticsearch
- Programme : <https://github.com/plawson/oc-projet3/tree/master/bitcoin-consumers>

- Restitution

- Dashboard Kibana

Implémentation de la solution

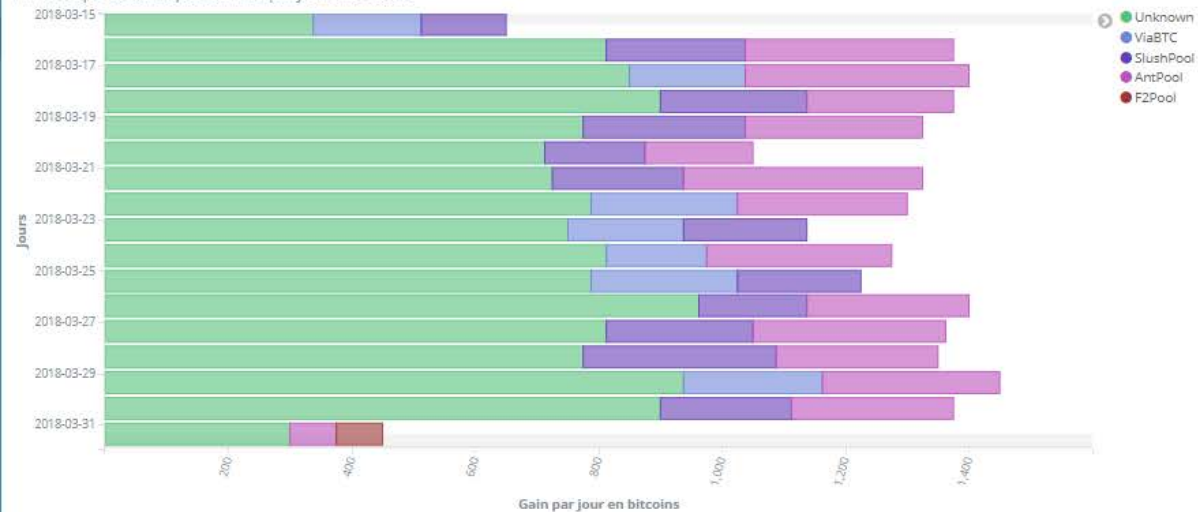


Graphes Bitcoins

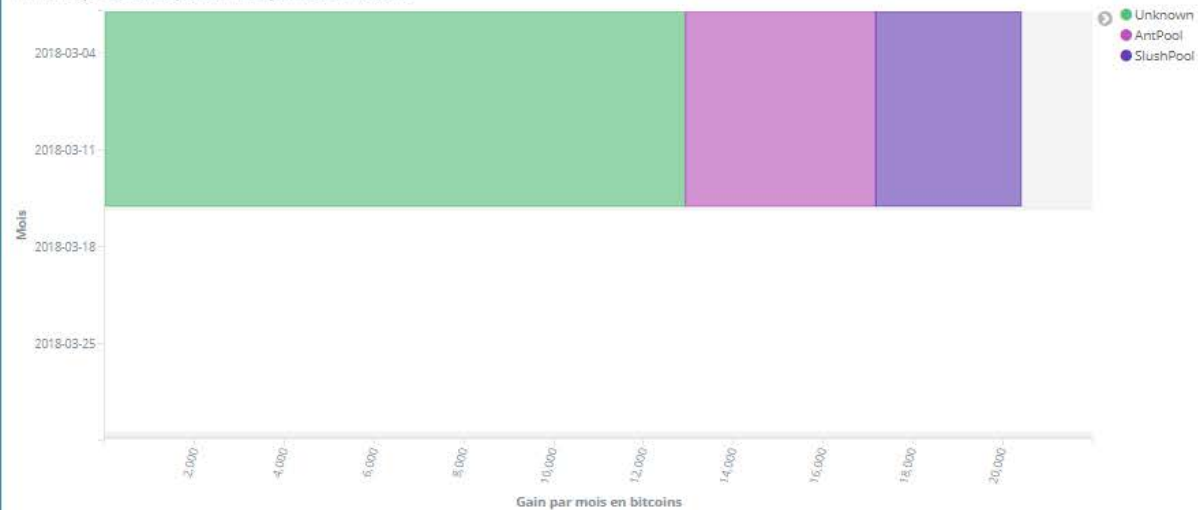


Graphes Mineurs

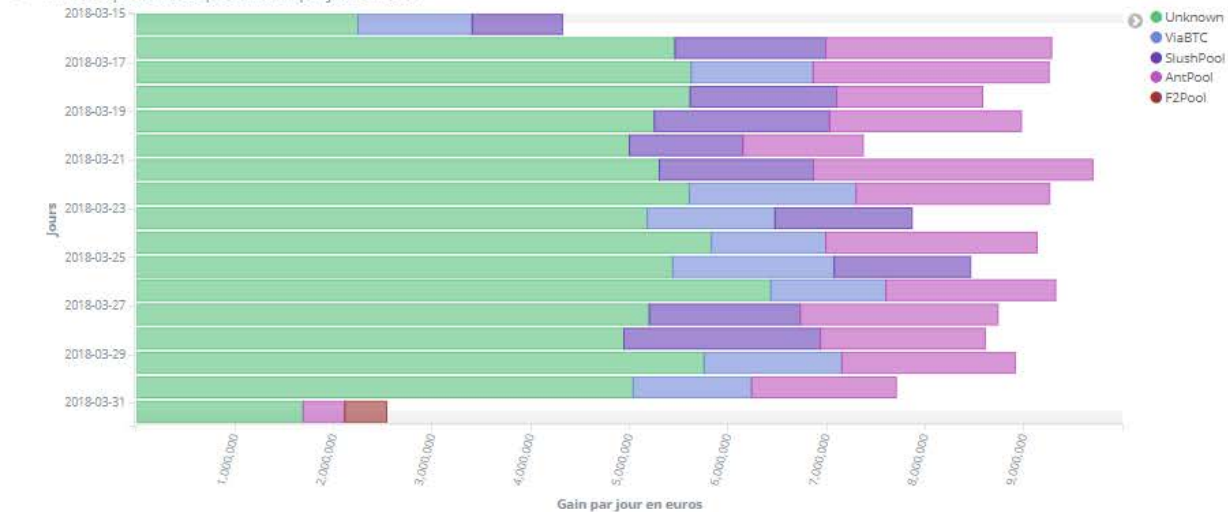
Mineurs qui se sont le plus enrichis par jour en bitcoins



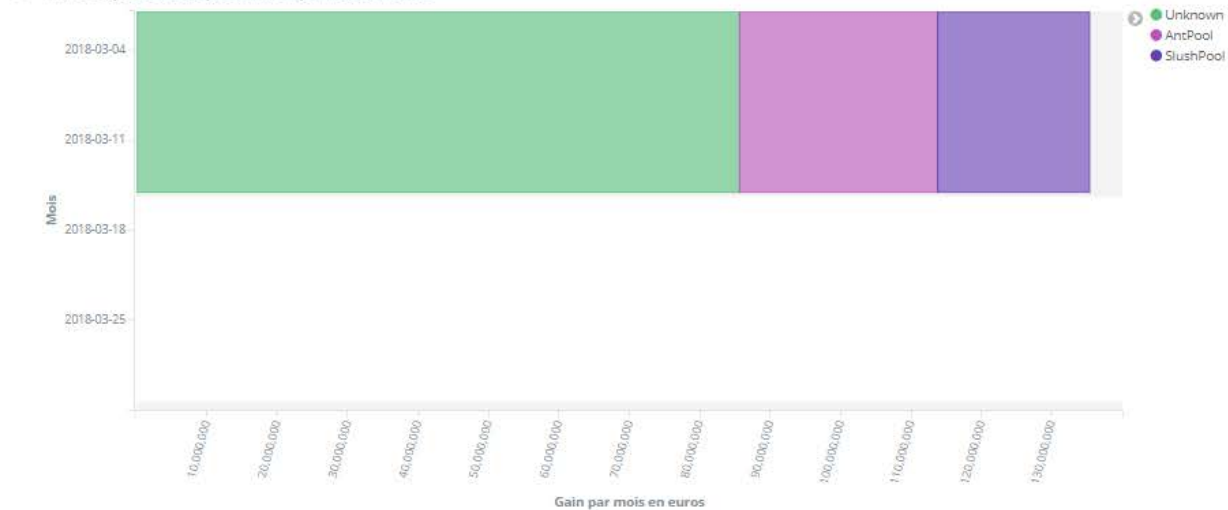
Mineurs qui se sont le plus enrichis par mois en bitcoins



Mineurs qui se sont le plus enrichis par jour en euros



Mineurs qui se sont le plus enrichis par mois en euros



Métriques Kafka

Consumers			
Show	10 ▾	entries	Search: <input type="text"/>
Consumer	↕ Type	↕ Topics it consumes from	↕
bpi-consumers	KF	bpi_eur : (0% coverage, 0 lag)	
btc-blk-consumers	KF	btc_blk : (0% coverage, 0 lag)	
btc-tx-consumers	KF	btc_tx : (0% coverage, 41 lag)	

Topics

Show entries

Search:

Topic	# Partitions	# Brokers	Brokers Spread %	Brokers Skew %	Brokers Leader Skew %	# Replicas	Under Replicated %	Producer Message/Sec	Summed Recent Offsets
__consumer_offsets	50	3	100	0	0	3	0	0.33	52,169
bpi_eur	1	3	100	0	0	3	0	0.03	2,493
btc_blk	1	3	100	0	0	3	0	0.00	235
btc_tx	10	3	100	0	0	3	0	2.00	349,767

Showing 1 to 4 of 4 entries

Previous **1** Next

Métriques Storm

Topology summary

Name	Id	Owner	Status	Uptime	Num workers	Num executors	Num tasks	Replication count	Assigned Mem (MB)
Bitcoin-Transactions	Bitcoin-Transactions-1-1525960127	root	ACTIVE	1d 13h 22m 39s	3	24	34	1	2496

Kafka Spouts Lag

Id	Topic	Partition	Latest Offset	Spout Committed Offset	Lag
btc-tx-spout	btc_tx	0	32882	32874	8
btc-tx-spout	btc_tx	1	32246	32241	5
btc-tx-spout	btc_tx	2	32266	32260	6
btc-tx-spout	btc_tx	3	32367	32362	5
btc-tx-spout	btc_tx	4	32326	32323	3
btc-tx-spout	btc_tx	5	32471	32464	7
btc-tx-spout	btc_tx	6	32536	32532	4
btc-tx-spout	btc_tx	7	32410	32407	3
btc-tx-spout	btc_tx	8	32203	32199	4
btc-tx-spout	btc_tx	9	32434	32429	5
bpi-spout	bpi_eur	0	2238	2238	0
btc-blk-spout	btc_blk	0	208	208	0

Spouts (All time)

Search:

Id	Executors	Tasks	Emitted	Transferred	Complete latency (ms)	Acked	Failed	Error Host	Error Port	Last error	Error Time
bpi-spout	1	1	2120	25440	4.723	2240	0				
btc-blk-spout	1	1	220	220	5.200	200	0				
btc-tx-spout	5	5	322220	322220	3.075	323900	0				

Showing 1 to 3 of 3 entries

Bolts (All time)

Search:

Id	Executors	Tasks	Emitted	Transferred	Capacity (last 10m)	Execute latency (ms)	Executed	Process latency (ms)	Acked	Failed	Error Host	Error Port	Last error	Error Time
bpi-es-bolt	1	1	0	0	0.009	506.634	2240	421.741	2240	0				
bpi-parsing-bolt	1	1	2000	2000	0.000	0.170	2240	0.152	2240	0				
btc-blk-es-bolt	1	1	0	0	0.006	522.091	220	1252.000	220	0				
btc-blk-parsing-bolt	1	1	260	260	0.000	0.156	2440	0.138	2440	0				
btc-tx-es-bolt	5	10	0	0	0.289	482.502	323900	476.808	323900	0				
btc-tx-parsing-bolt	5	10	323880	323880	0.000	0.121	346260	0.114	346320	0				

Showing 1 to 6 of 6 entries

Scénario catastrophe

- Composants techniques (Zookeeper, kafka, storm, Elasticsearch...) :
 - La perte d'une instance est gérée par la configuration des ressources k8s.
 - Si une instance stateful (ex: Kafka) tombe, elle est redémarrée sur un autre nœud et le filesystem de l'instance perdue est automatiquement remontée sur le nouveau pod et sera accédé par le nouveau conteneur sans perte de données.
 - En cas de perte d'une instance stateless (ex: Master node ES), elle est automatiquement redémarrée sur un autre nœud.
- Défaillance filesystem:
 - Dans l'environnement d'intégration, les tests ont été réalisés avec des disques hébergés sur un NAS (4 NIC) dans un volume en RAID 6. Ils sont montés par ISCSI via Disk Mapper Multipath (DM-Multipath) sur la machine hôte. En cas de corruption d'un ou deux disque(s), il suffit de le(s) remplacer.