Technical note

As the technical excellence will be a determining factor we felt we had to add this technical note to our application. We had the chance to be involved in the testnet since day 1 and to have a node always in the Authority Set (both Audit and Leader, currently Leader). @luap has been following and involved in every single tech discussion around testnet that happened on Slack and Discord.

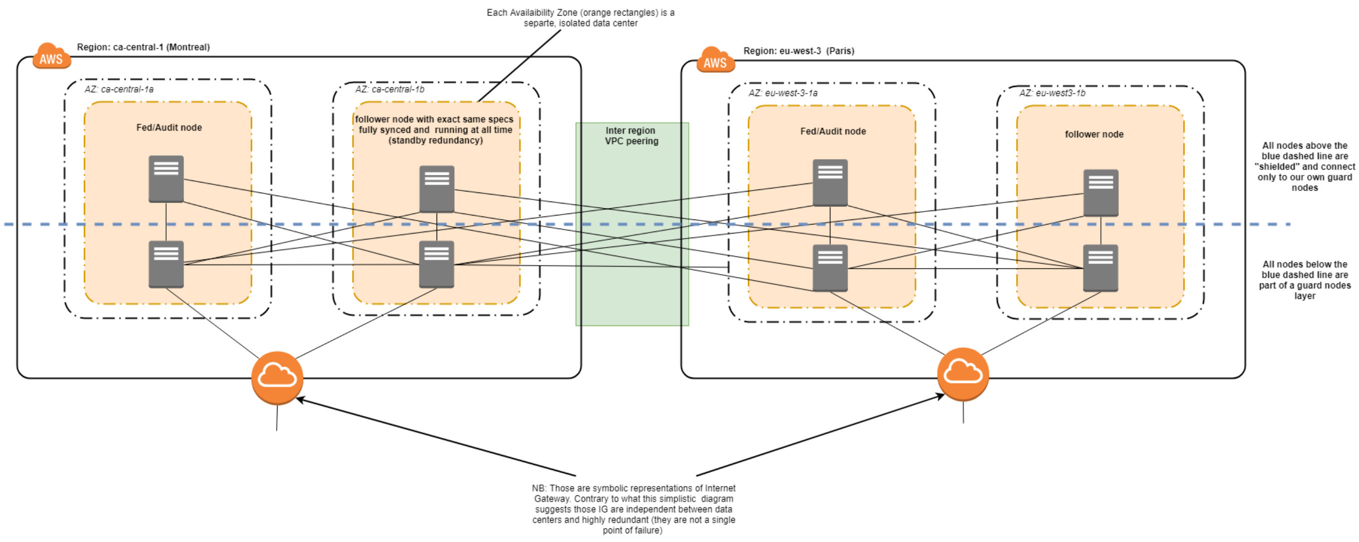
Hosting platform

We want to use AWS cloud provider in Canada for the following reasons:

- Paul has been Software Engineer at Amazon for over 3 years using AWS in his daily job.
- In addition, Paul is about to become AWS Solution Architect Associate certified proving extensive knowledge of the ins and outs and good practices of the platform.
- AWS offers extensive flexibility in the design of our architecture and offer advanced capabilities. We are going far further than just spinning up an instance as you can see below.
- AWS EC2 has a SLA of 99.99% uptime.
- We are located and will be operating in Canada so AWS Canada region is the most sensible. We picked the second area in Paris to have servers on two different continents.

Architecture

This is our ideal architecture with 2 authority nodes spread on 2 continents and protected by guards:


Each Availaibility Zone (orange rectangles) is a separte, isolated data center
AWS
Region: ca-central-1 (Montreal)
AZ: ca-central-1a
Fed/Audit node
AZ: ca-central-1b
follower node with exact same specs fully synced and running at all time (standby redundancy)
Inter region VPC peering
AWS
Region: eu-west-3 (Paris)
AZ: eu-west-3-1a
Fed/Audit node
AZ: eu-west3-1b
follower node
All nodes above the blue dashed line are "shielded" and connect only to our own guard nodes
All nodes below the blue dashed line are part of a guard nodes layer
NB: Those are symbolic representations of Internet Gateway. Contrary to what this simplistic diagram suggests those IG are independent between data centers and highly redundant (they are not a single point of failure)

Please note the use of inter-region VPC peering allowing both locations to be virtually part of the same network (without the need to use a VPN or anything else).

Usage of guards

We have demonstrated our ability to set up a functional guard configuration as you can see here: http://52.202.51.229:8090 Please note that this node is a *leader*: we managed to set up this configuration while keeping our status of leader at all time, demonstrating at this occasion that we can handle brainswaps operations perfectly.

Hosts config

AWS EC2 offers a wide variety of instance types that gives us a great deal of flexibility to adapt to the evolving requirements. We'll start by using the new M5 family (m5.xlarge) as it has strong and balanced specs with many optimizations.

Few other points we would like to highlight about our config:

- The blockchain itself is stored on a separate partition and volume from the boot volume.
- We are using LVM (Logical Volume Manager) volumes on the testnet. This allows us to virtually have no limit in our disk space to store the growing blockchain.
- We'll use SSD volumes with guaranteed high IOPS performance (EBS io1).

On call

We'll be taking care of the maintenance and responding incidents ourselves. This removes any intermediary, we have a direct knowledge of our architecture and the software we are running. That gives us the additional advantage of not having any cost regarding sysadmin/devops allowing us to be cost efficient. Both Paul and Lucia will be point of contact and be notified in case of incident, increasing our reliability. Paul has 3 years of experience of 24/7 on call for Amazon (for 1 week every couple of weeks) where the maximum time to come online is 15 minutes.

Monitoring and alarm

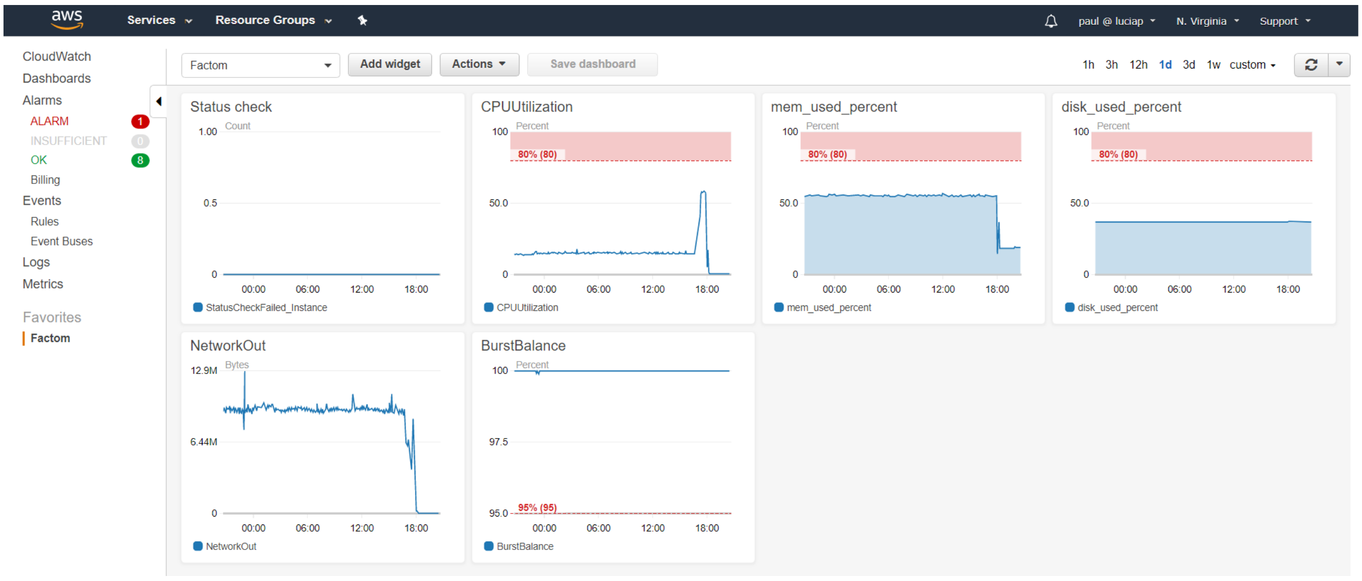
First, we have implemented 3 layers of monitoring. We collect:

- Nitro hypervisor metrics: CPU, memory, network bandwidth...
- In-guest metrics (CloudWatch agent): disk space...
- Application metrics: leveraging embedded prometheus + grafana (for size of message queues for instance) + monitor factomd process (in house tool). In the future we wish to transmit those metrics to CloudWatch to have a unified solution.

Those metrics are used by CloudWatch alarms that will trigger if any anomaly is detected. Those alarms immediately trigger an email and a SMS to all members of LUCIAP (via SNS). A pager application installed on our mobile phone will make sure we don't miss those signals.

If you look at the testnet chat history you can see that on multiple occasions @luap was the first to call out failure of the testnet: this is not black magic, it's just proper alarming in place, even on testnet.

Here's a glimpse to our simple testnet dashboard monitor to control the health of our node at a high level:



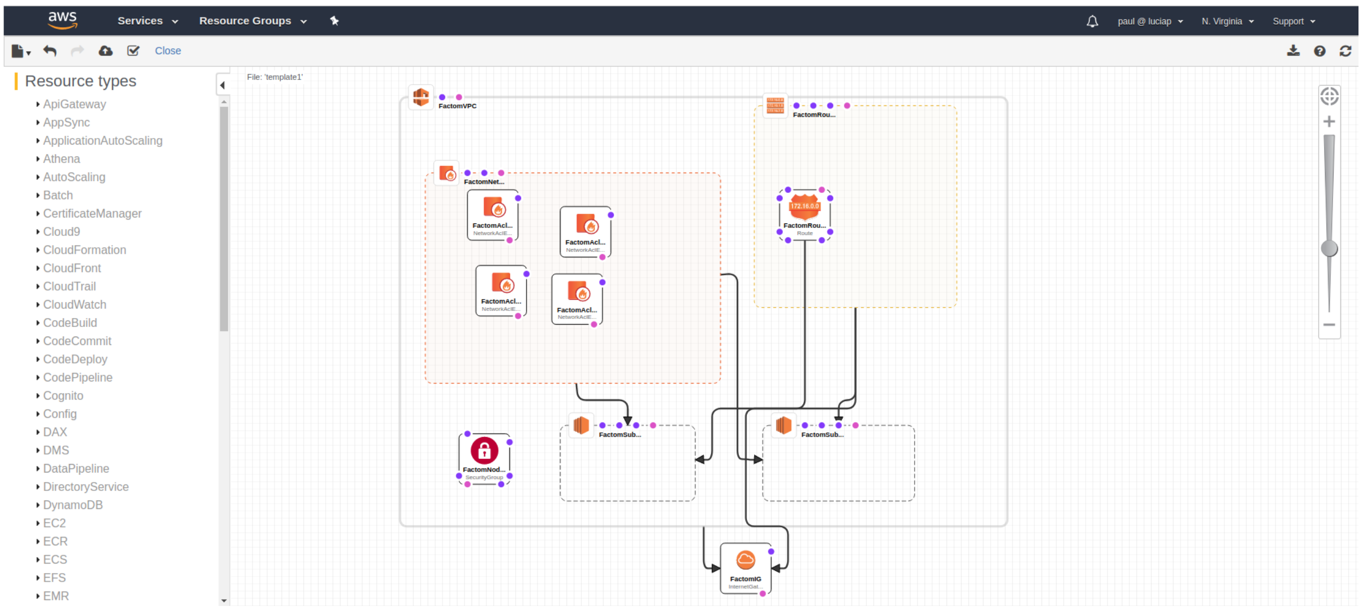
Infrastructure as code

We have fully embrace the Infrastructure as Code (IaC) principle. AWS Cloudformation allows us to define our previously describe architecture as code. The benefits are tremendous:

- We can redeploy our entire infrastructure in any AWS region in minutes (we are excluding the boot time in this statement).
- We considerably reduce variability of what is deployed and human mistakes.
- Automatically set up monitors and alarms that can be easily overlooked if done manually.
- We can spawn pre-configured/bootstrapped Factom node on demand according to a specific role (authority, guard, follower).

Our CloudFormation templates are pillars of our recovery plan (for the reasons mentioned above). We have already developed a set of Cloudformation templates. Those are effectively hours of work to be fully functional and polished and that's why we won't be sharing them publicly. We believe that those Cloudformation templates give us a competitive advantage over other candidates.

We can show you a preview of one of our infrastructure template in the visual editor:



Security

Securing our infrastructure, identities and funds are of paramount importance.

Infrastructure

As we would be operating on AWS we would have a shared responsability model, meaning AWS is responsible for the security *of* the cloud (infrastructure, hardware...) while we are responsible for the security *on* the cloud (guest OS, application...).

Here are some highlights of measures we have taken:

- Our AWS account is dedicated entirely to our Factom operations and only accessible by Paul and Lucia
- Access to it is protected by Multi Factor Authentication (MFA)
- We leverage all the networking tools offered by AWS: we have created a specifically designed network (VPC) with proper network ACLs together with strict firewall rules (Security groups). The surface of attack is minimal
- Access to host is only possible using public-key cryptography (no password). SSH is on non-standard port and external firewall allows only restricted set of IPs to connect. We are considering MFA for access but honestly what's already in place is quite secure...
- We monitor the updates of our fleet through AWS Systems Manager

Identifies and funds

In the short term we plan to store private keys of our identities and wallet on multiple offline encrypted volumes spread in multiple locations. We will be seeking more long-term solutions (that maybe we should discuss within the community). We are also hoping to use Ledger once compatible with FCT.

Operational excellence

We strive toward operational excellence and to do so we'll be putting in place multiple processes:

- We have set up an internal wiki with Standard Operation Procedures (SOPs) of common task to reduce human mistakes (how to handle updates in our architecture, how to perform node bootstrapping, how to use our CloudFormation templates...)
- Regular review of our internal documentation: up to date architecture document...
- Monthly Operational Readiness Reviews (ORRs): includes failure testing and recovery, brain transfer, deployment of CloudFormation templates, test notification system...
- Weekly review of metrics and incident that happened. Corrective measures.
- Weekly snapshot of the volume containing the blockchain (taken from a follower node) to allow "fast" bootstrap in case of emergency (successfully tested).

Additional tools

It is in our intention to develop some tools for safe brainswapping. As of today the manual swapping is extremely human error prone. We know this is the concern of a few other participants too and will be willing to join effort.