kindred

OpenStack compliance and security

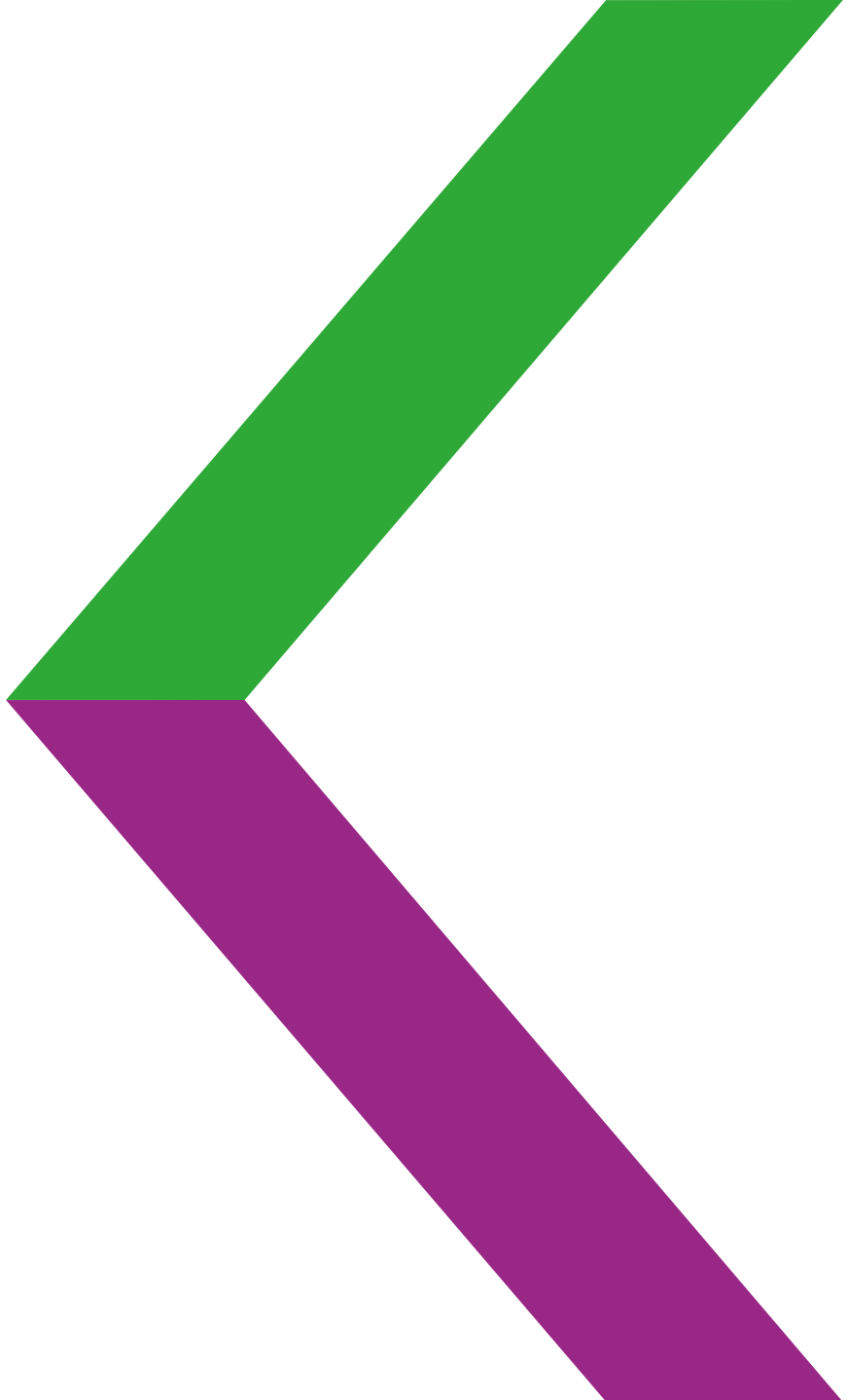
Antonio Pezo / Gauvain Pocentek

2020/01/22

Agenda

- A bit of context
- Availability of resources
- Compliance and security
- Traceability and auditing
- Questions

A bit of context

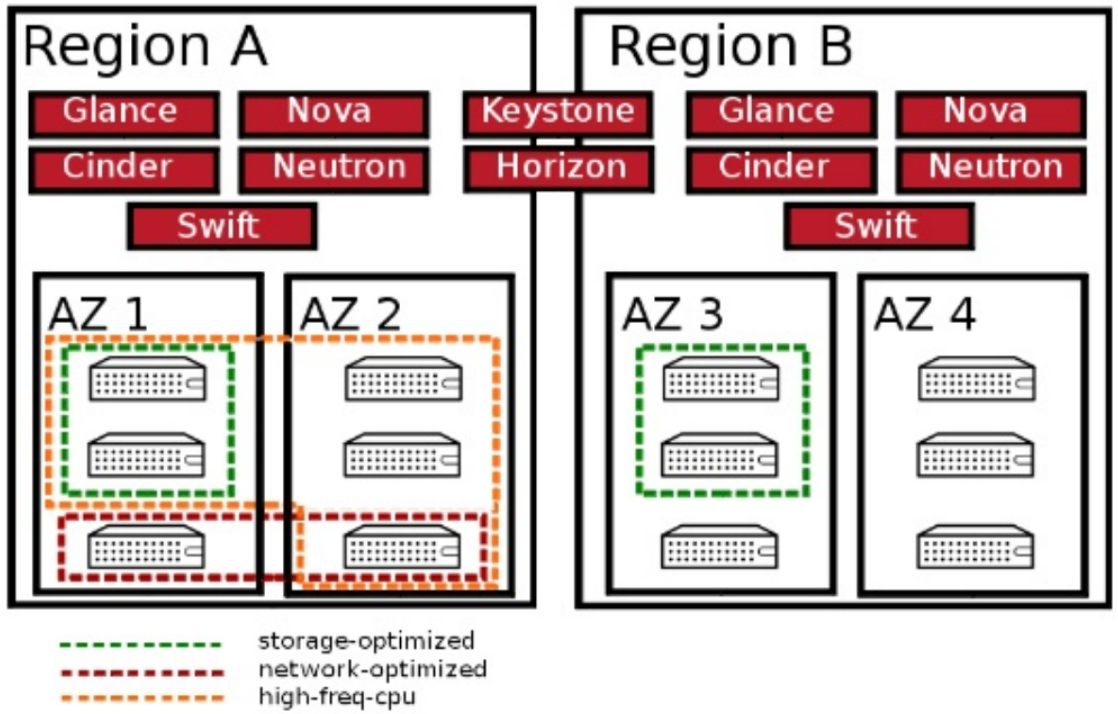


- As any tech company we care about security to protect our users and ourselves
- Our line of business implies a lot of compliance rules:
  - ISO 27001
  - Gambling regulations
  - PCI-DSS environments
- Using a cloud platform changes the way compute resources are managed
  - More users can create resources
  - We need the same level of security we have for legacy infrastructure
  - Although cloud native apps are the focus, we need to take care of legacy systems in the cloud as well
- OpenStack provides the features we need, but making it comply with our standards requires a bit of tuning!

Availability

Architecture options

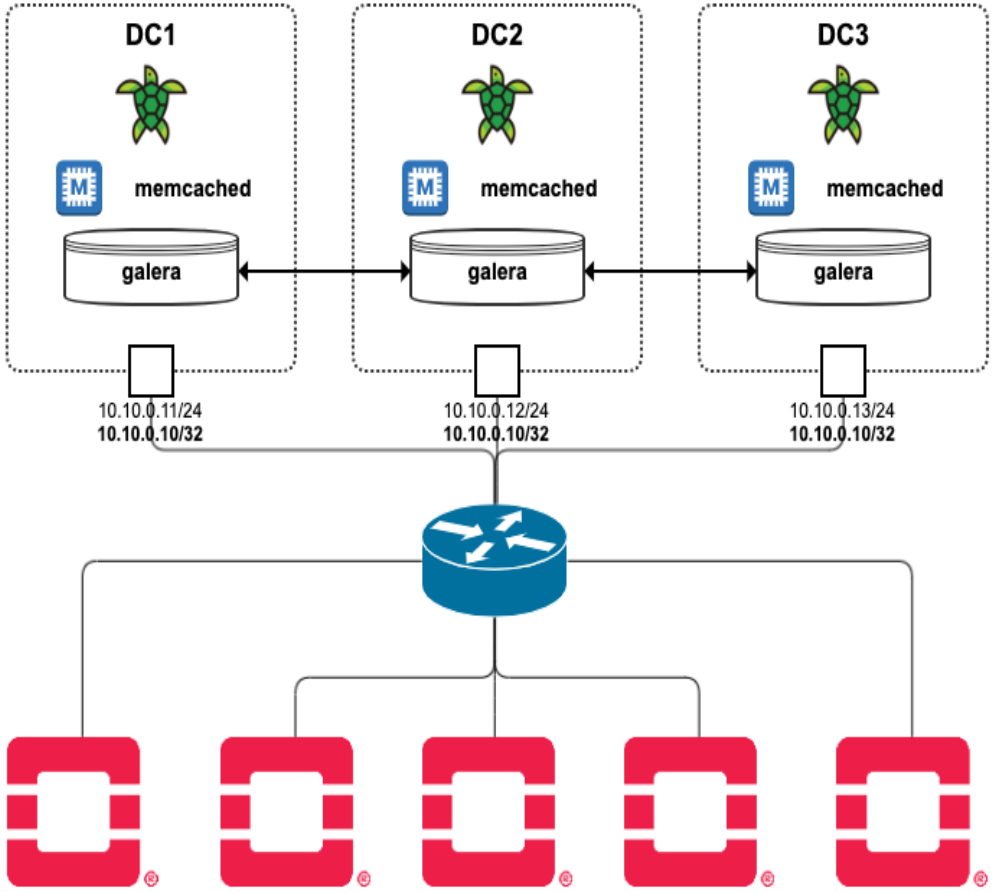
- How do you make sure that compute resources are always available?
- OpenStack provides multiple options:
  - Availability zones
  - Regions
- We want to survive the loss of a DC: multi-region is the best option



Focus on multi-region

- The theory: Keystone shared by multiple OpenStack clusters
- An OpenStack region (DC) can disappear, the other regions are still available
- But:
  - If Keystone goes down, all the APIs are down
  - How do we make sure that Keystone stays available?
- A solution:
  - Keystone available in multiple DC
  - MariaDB / Galera database stretched across on multiple DC as well
  - Anycast / BGP to reach the closest available Keystone

Focus on multi-region

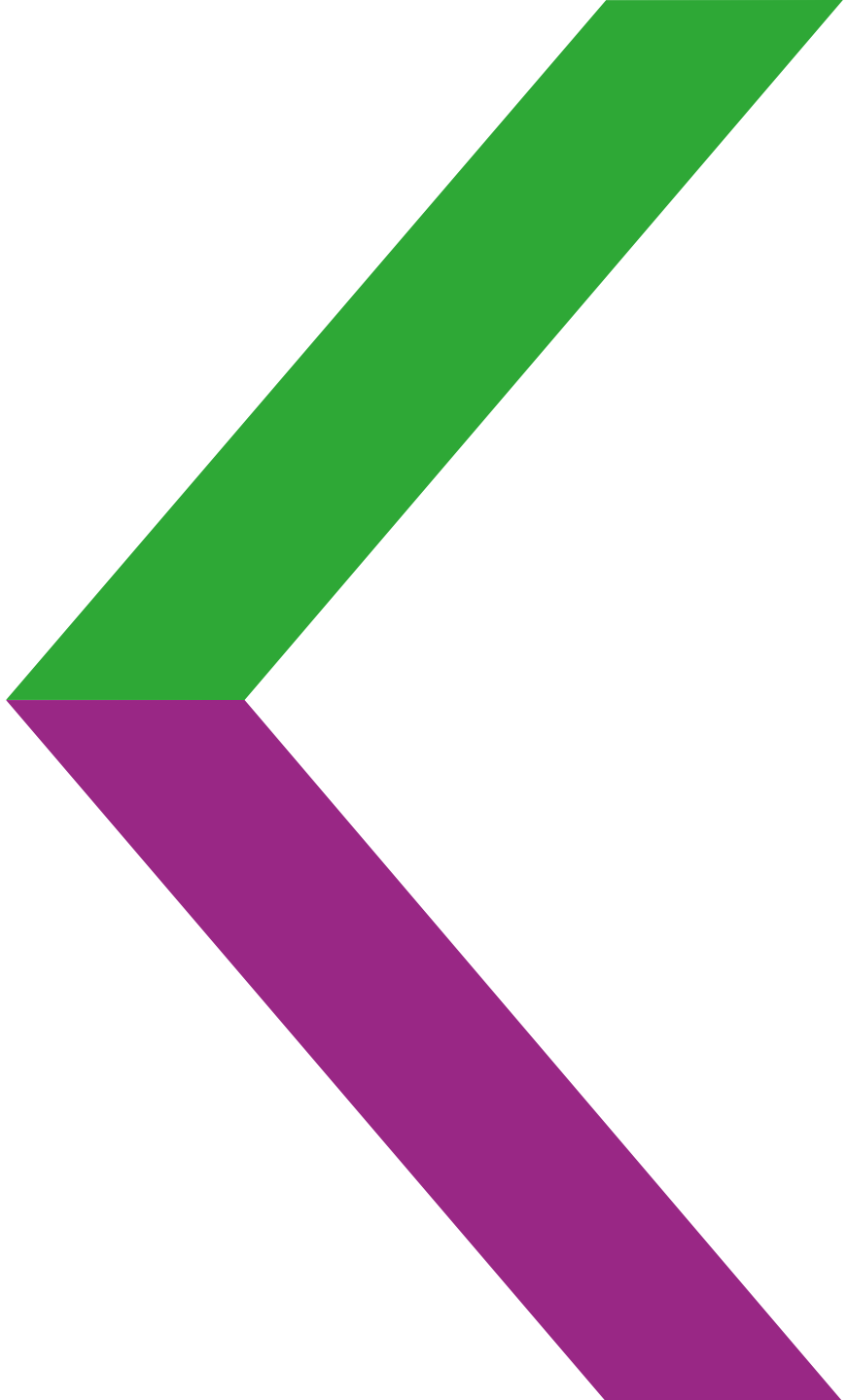


- Very few writes to the DB: latency doesn't hurt
- Keystone talks to the local Galera and to the local Memcached
- ExaBGP used to monitor the services and handle the BGP protocol: quick and automated failover

Monitoring

- One tool to monitor everything is not realistic
- What can be used:
  - "Legacy" monitoring for system-related checks (running processes, disk status, CPU/RAM)
  - Prometheus to monitor capacity: multiple OpenStack exporters are available
  - Logs centralization and alerts on specific events (RabbitMQ failures, increase of 500 errors, and so on)
- Something to look at: Vitrage

Compliance and Security



Authentication requirements

- Password policy and rotation:
  - Handled by AD/LDAP for users
  - Needs to be handled in Keystone for service accounts stored in DB

- Keystone provides a set of options for PCI-DSS compliance, for example:
  - Password strength
  - Lockout on multiple authentication failures
  - Password expiration

Authentication requirements

- In practice it can become a nightmare for operations:
  - Changing the password in Keystone breaks the authentication for applications
  - Need to track the expiration date of passwords

- This also applies for OpenStack components themselves:
  - Password rotation implies downtime

- Application credentials to the rescue:
  - Additional key/secret pairs that can be used to authenticate a user
  - App credentials can have less privileges than their associated user
  - Multiple sets of credentials can exist at the same time: rotation becomes easy
  - User / service account is only used to create new credentials
    - Its password can be changed often without impacting applications

Virtual networks

- Firewall policies managed by the network and security teams
  - Giving access to the whole Neutron API for users can lead to security risks
  - Networks can be owned by the OpenStack admin project and shared with tenants:
    - Prod and non-prod (dev/QA) networks are never available in the same tenant
    - Specific networks can be made available to specific tenants or shared between tenants

- Implementation:
  - Neutron RBACs: no shared networks, whitelisting
  - Networks and subnets quotas set to 0 by default
  - Custom API policies for some features

Virtual networks

- Other network configurations:
  - No NAT, makes easier to track IPs of services
    - So no floating IPs
  - Virtual routers still in use though:
    - Only 1 IP available on the “floating IP networks”: the router gateway
    - VXLAN networks (“tenant networks”) provide nice features:
      - Easier live migration
      - Completely internal to OpenStack, no external config needed
  - Provider networks required, mainly to support multicast

Traceability and auditing

OpenStack API logs

- API requests logging in OpenStack is not optimal:
  - No consistency across components, for example:
    - Single line with all the information in neutron-server.log
    - 3 lines with a common ID field in cinder-api.log
  - Information can be scattered across multiple files:
    - Httpd logs for IP-related information
    - <service>.log for authentication (Keystone) information
  - IP logging is not working out of the box
    - Who cares about the proxy IP address? We want the user IP!

Some solutions

- Patching of OpenStack services:
  - IP information in the application logs can be easily added with a few lines of code
    - This implies additional maintenance
    - Unless pushed upstream

- CADF (Cloud Auditing Data Federation)
  - Provides standardized json log for each API call
  - Implemented in most OpenStack components
  - Requires additional configuration

Auditing

- Virtual resources creation completely automated and going through processes

- We trust our users, but things can still happen outside of the process, so we track:
  - User/group assignments to projects
  - Network assignments (RBACs)
  - Instance ownership and network connectivity

Questions?

Get in touch with Kindred:

Mail: claire.skorupa@kindredgroup.com

Phone: +46 70 875 36 86