



삼성 오픈소스 컨퍼런스
SAMSUNG OPEN SOURCE CONFERENCE

OPEN YOUR UNIVERSE WITH SOSCON

오픈스택을 통한 내 성장기

—
카카오 클라우드 컴퓨팅셀
송지현

2015.10.28

들어가기 앞서...



“저”에 대한 이야기고요

삼성 오픈소스 컨퍼런스
SAMSUNG OPEN SOURCE CONFERENCE



<http://blogs.disney.com/oh-my-disney/2014/04/28/15-things-you-thought-while-you-watched-the-lion-king/>

사고하는 방식에 대해서 봐주시면 좋을 것 같아요

삼성 오픈소스 컨퍼런스
SAMSUNG OPEN SOURCE CONFERENCE



다시 한번 소개하겠습니다.

삼성 오픈소스 컨퍼런스
SAMSUNG OPEN SOURCE CONFERENCE

kakao



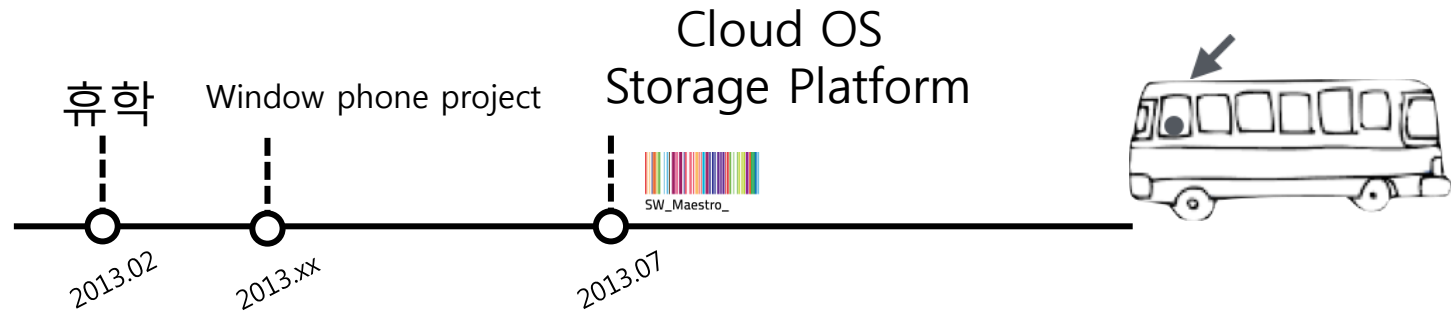
SW_Maestro_



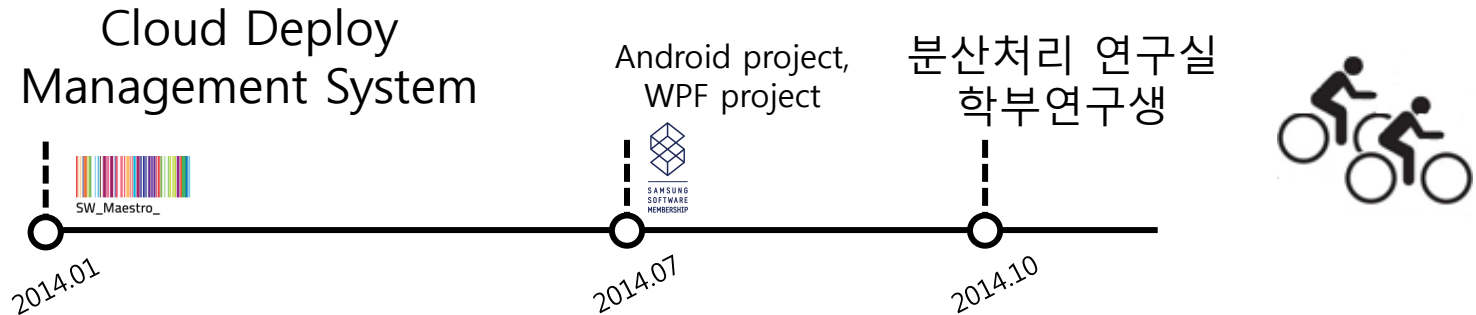
SAMSUNG
SOFTWARE
MEMBERSHIP



2013년도



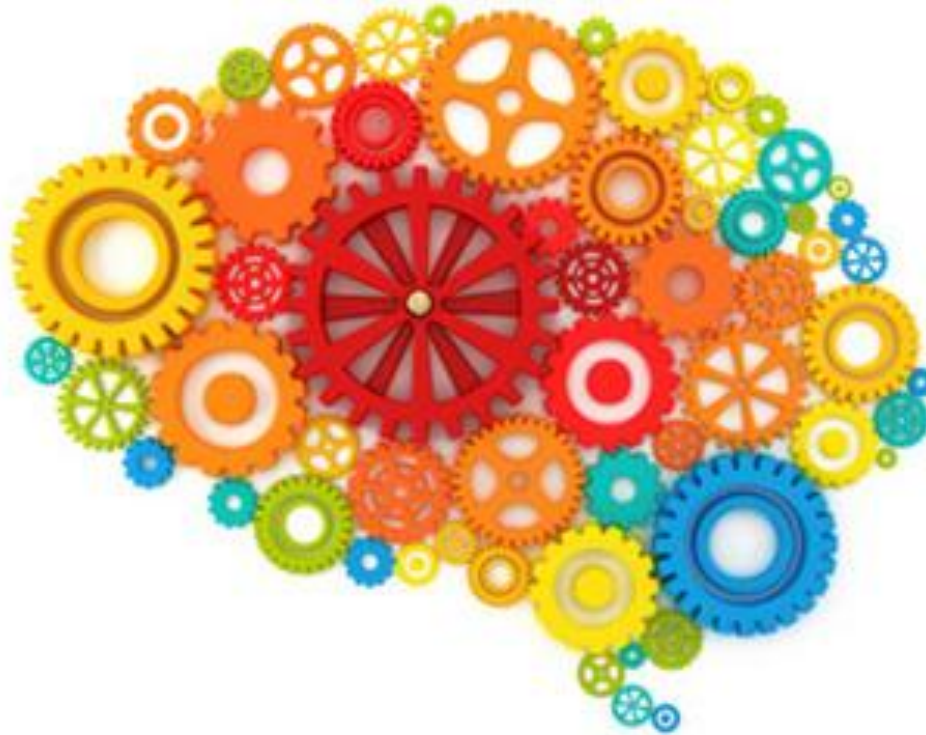
2014년도



2015년도



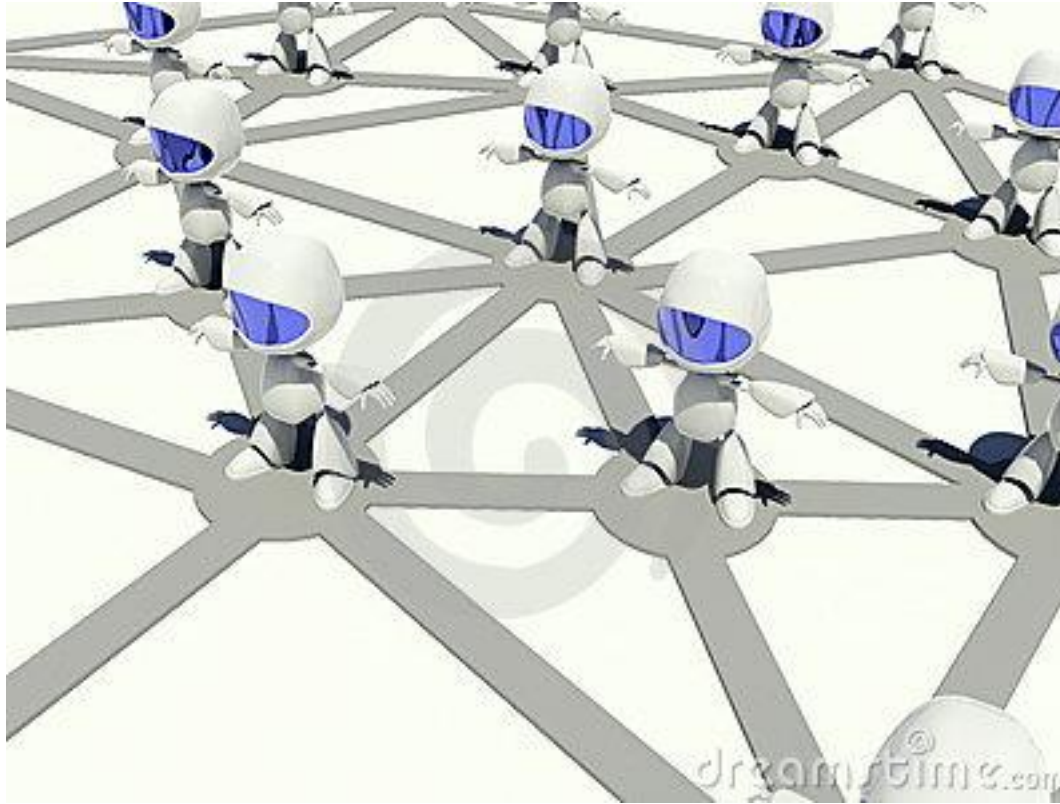
그래서 사실...



<http://www.corenxos.com/virtual-labs/>

어떻게든 동작하기만 하면 되는... 프로토타입 개발?
기초 수준으로 사용한 언어도 다양함...





클라우드나 네트워크, 분산 & 병렬처리 등에 관심이 많음(지금도!)



SW_Maestro_

의 첫 프로젝트 시작은...

멘토분들이 각자 해보고 싶은 주제를 발표하고...

멘티가 신청하는 방식이었습니다.

하드웨어단부터 서비스, 클라우드까지... 다양한 주제의 프로젝트가 있었어요.

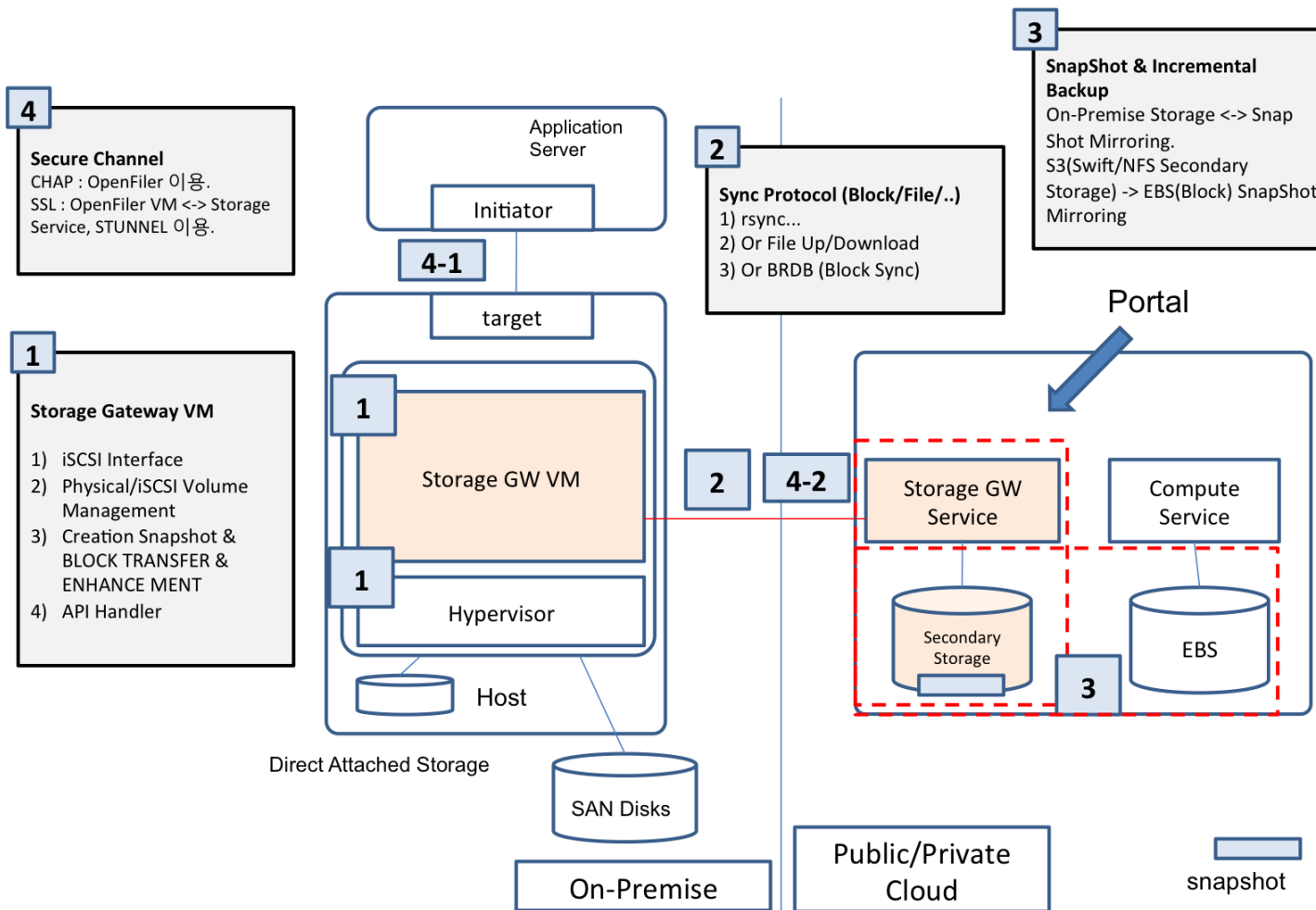
근데 거기서 그만...

꿈만 꾸던 프로젝트를 봐버렸어요



출처 : 에비쥬

1차 프로젝트에서 우리 멘토님이 보여주신...



HYBRID CLOUD STORAGE SERVICE PLATFORM...

당시에 뭘진 모르겠지만

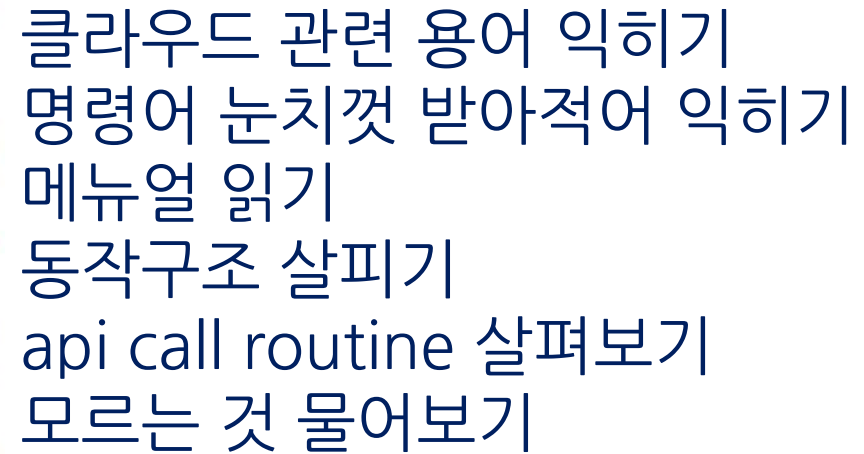
1. PRIVATE CLOUD 환경을 구성하고,
2. 분산 스토리지를 구성하고,
3. 거기에 무언가 스토리지를 자동으로 백업하고, 복구하고,

등등등... 을 그린 프로젝트였어요.

그 프로젝트 너무너무 참여하고 싶어서...

삼성 오픈소스 컨퍼런스
SAMSUNG OPEN SOURCE CONFERENCE





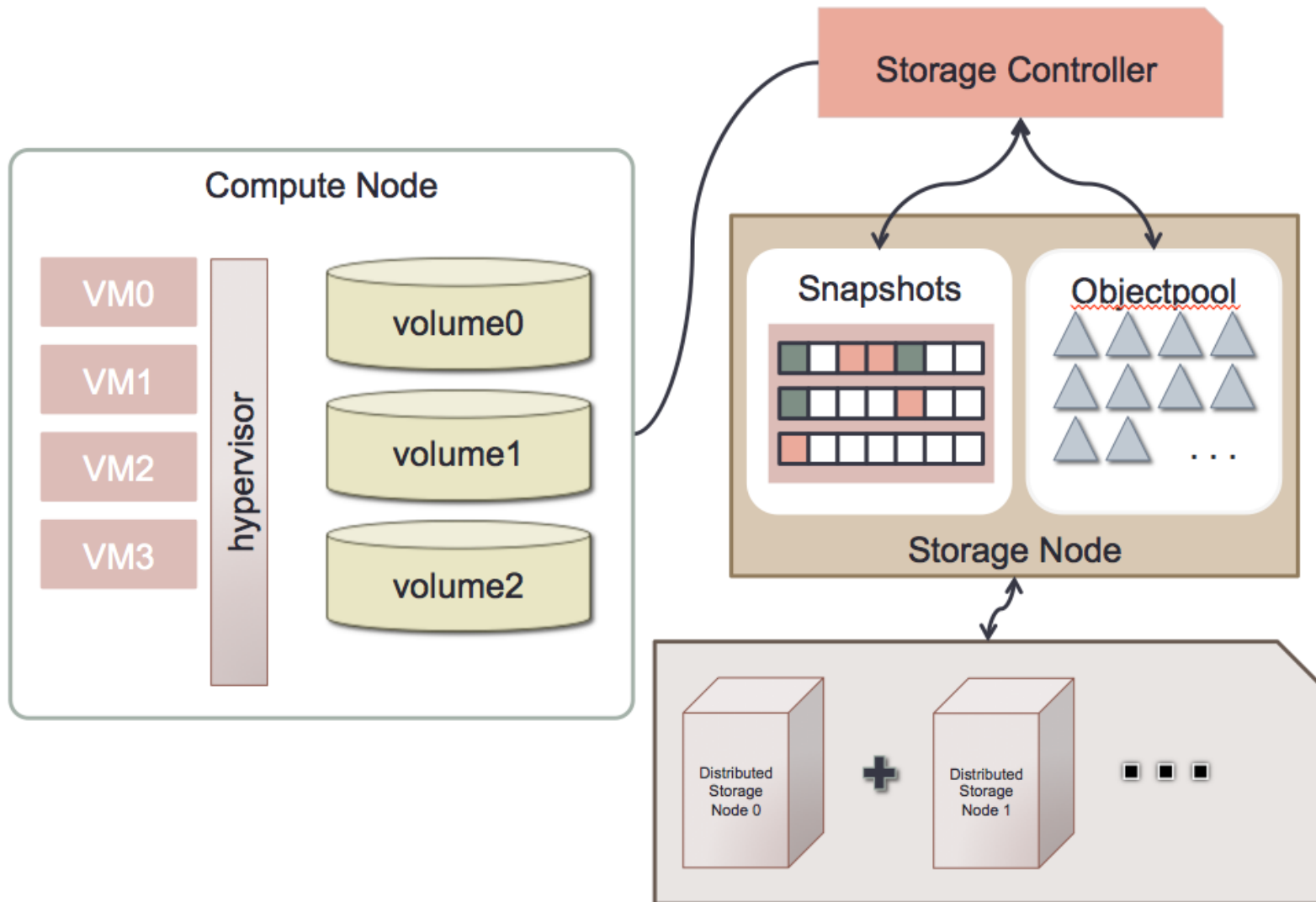
그냥... 해보라는 것부터 하나부터 열까지 차근차근 정리

가장 도움이 되었던건 남은 로그!!

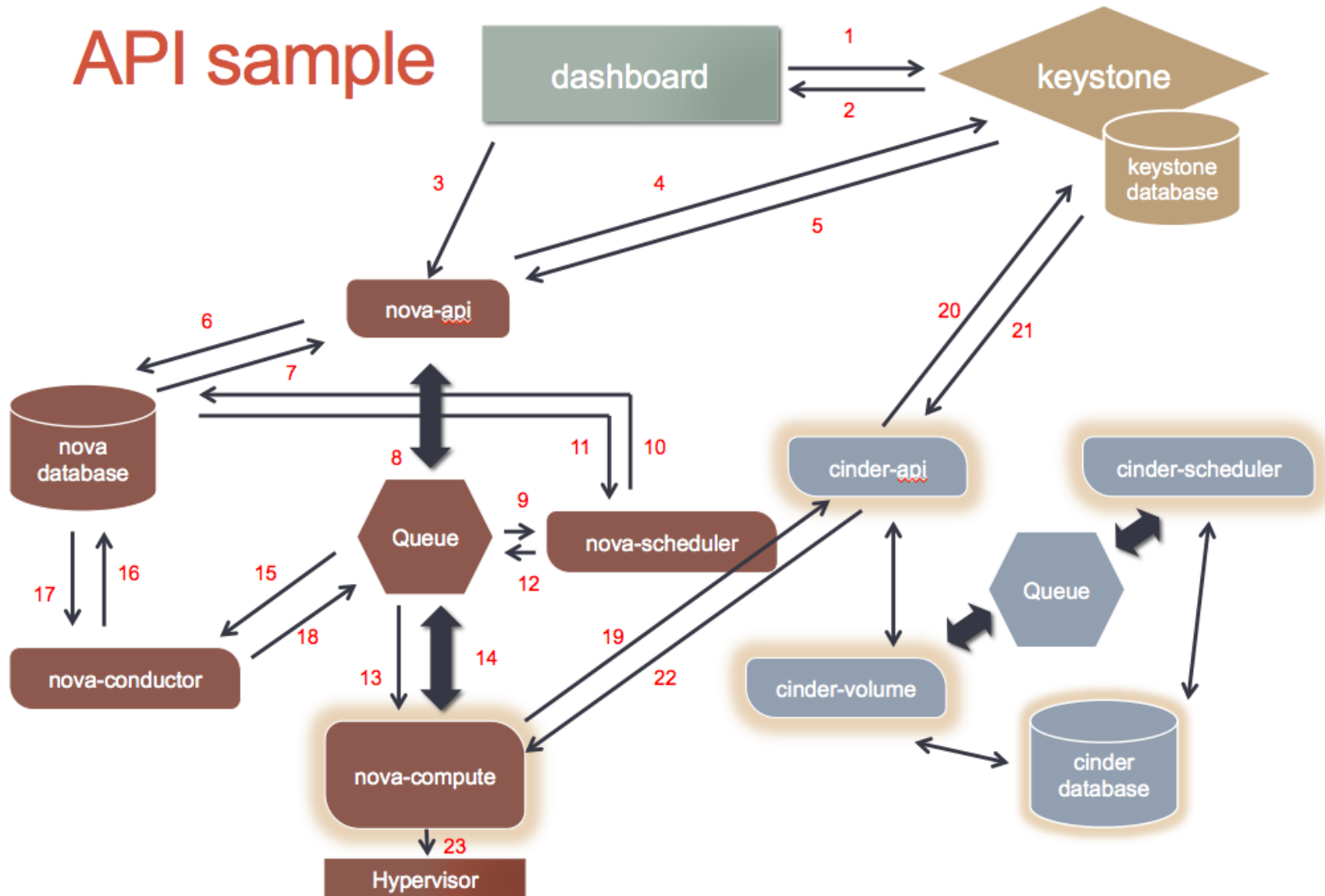
```
f = open('/opt/stack/logs/xxx_api.py.log', 'a')  
f.write("log 1 of init_instance "  
       "in /opt/stack/nova/nova/compute/manager.py")  
f.close() ## DEBUG BY JIHYUN FOR SEARCH
```

이런식의 파일 입출력을 통해.. (LOG.debug도 모르고...)

하나씩 하나씩 정리



API sample



```
Jenny@JiHyunSonguiMBP:~/git/SOMA2014/CDMS/cinder/cinder ((detached from 2014.1))
$ ls
__init__.py      context.py      image           policy.pyc      scheduler       units.py        wsgi.py
__init__.pyc     context.pyc     keymgr          quota.py        service.py      units.pyc       wsgi.pyc
api              db              locale          quota.pyc       service.pyc     utils.py        zonemanager
backup           exception.py    manager.py      quota_utils.py  test.py         utils.pyc
brick           exception.pyc  manager.pyc     quota_utils.pyc testing          version.py
common          flow_utils.py  openstack       rpc.py          tests           version.pyc
compute         flow_utils.pyc policy.py        rpc.pyc         transfer        volume

Jenny@JiHyunSonguiMBP:~/git/SOMA2014/CDMS/cinder/cinder ((detached from 2014.1))
$ grep -r "def _create" . --exclude-dir=tests
./api/contrib/types_manage.py:    def _create(self, req, body):
./api/openstack/wsgi.py:    def _create_link_nodes(self, xml_doc, links):
./api/versions.py:    def _create_feed(self, versions, feed_title, feed_id):
./api/versions.py:    def _create_version_entry(self, version):
./backup/drivers/ceph.py:    def _create_base_image(self, name, size, rados_client):
./backup/drivers/swift.py:    def _create_container(self, context, backup):
./backup/drivers/tsm.py:    def _create_unique_device_link(backup_id, volume_path, volume_id, bckup_mode):
./brick/local_dev/lvm.py:    def _create_vg(self, pv_list):
./image/glance.py:    def _create_glance_client(context, netloc, use_ssl,
./image/glance.py:    def _create_static_client(self, context, netloc, use_ssl, version):
./image/glance.py:    def _create_onetime_client(self, context, version):
./openstack/common/log.py:    def _create_logging_excepthook(product_name):
./service.py:    def _create_service_ref(self, context):
./volume/api.py:    def _create_snapshot(self, context,
./volume/drivers/glusterfs.py:    def _create_snapshot(self, snapshot):
./volume/drivers/glusterfs.py:    def _create_qcow2_snap_file(self, snapshot, backing_filename,
```

```
Jenny@JiHyunSonguiMBP:~/git/SOMA2014/CDMS/cinder/cinder ((detached from 2014.1))
$ ls
__init__.py      context.py      image           policy.pyc      scheduler       units.py        wsgi.py
__init__.pyc     context.pyc     keymgr          quota.pyc       service.py      units.pyc       wsgi.pyc
api              db              locale          quota.pyc       service.pyc     utils.py        zonemanager
backup           exception.py    manager.py      quota_utils.py  test.py         utils.pyc
brick            exception.pyc  manager.pyc     quota_utils.pyc testing          version.py
common           flow_utils.py  openstack       rpc.py          tests            version.pyc
compute         flow_utils.pyc policy.py        rpc.pyc         transfer         volume

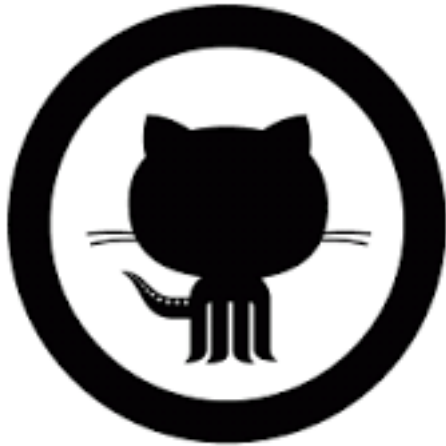
Jenny@JiHyunSonguiMBP:~/git/SOMA2014/CDMS/cinder/cinder ((detached from 2014.1))
$ grep -r "def _create" ./ --exclude-dir=tests
./api/contrib/types_manage.py: def _create(self, req, body):
./api/openstack/wsgi.py: def _create_link_nodes(self, xml_doc, links):
./api/versions.py: def _create_feed(self, versions, feed_title, feed_id):
./api/versions.py: def _create_version_entry(self, version):
./backup/drivers/iscsi.py: def _create_iscsi_image_link(self, req, body, backup_mode):
./backup/drivers/iscsi.py: def _create_iscsi_image_link(self, req, body, backup_mode):
./backup/drivers/tsm.py: def _create_tsm_image_link(self, req, body, backup_mode):
./brick/local_dev/lvm.py: def _create_lv(self, pv_list):
./image/glance.py: def _create_glance_image(self, context, req, body, version):
./image/glance.py: def _create_glance_image(self, context, req, body, version):
./image/glance.py: def _create_glance_image(self, context, req, body, version):
./openstack/common/log.py: def _create_log_entry(self, req, body):
./service.py: def _create_service_ref(self, context):
./volume/api.py: def _create_snapshot(self, context,
./volume/drivers/glusterfs.py: def _create_snapshot(self, snapshot):
./volume/drivers/glusterfs.py: def _create_qcow2_snap_file(self, snapshot, backing_filename,
```

엄청나게 각 계층별로 차근차근 함수를 호출하는데...

함수 명을 보면서 추론해 나아가다 보면

모듈의 역할이 분명해지고...

나름의 구조를 파악하기가 좋아요



<http://advisoranalyst.com/glablog/2015/08/28/when-to-deploy-capital.html>

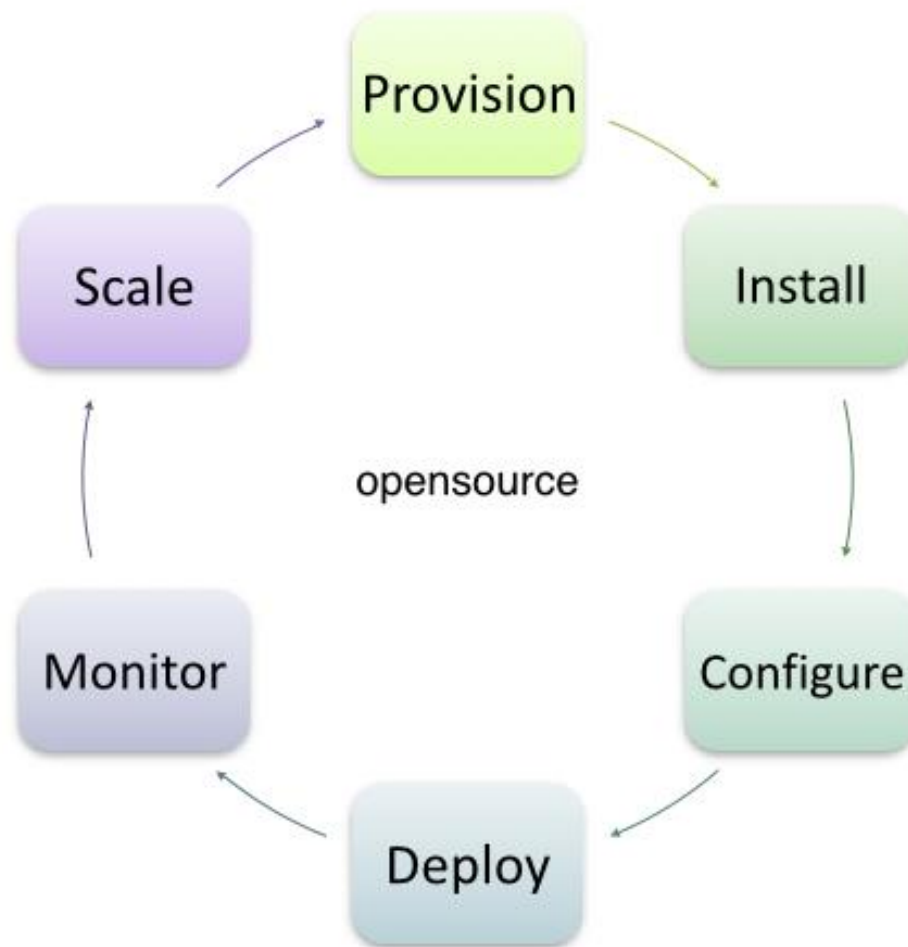




<http://egloos.zum.com/sadsong/v/849421>



<http://www.jeffbullas.com/2015/06/19/10-top-digital-marketing-automation-tools/>



1. Private Cloud 구축(이번엔 멀티노드로!)
2. 클라우드 내 개발 환경/Application 배포시스템
3. Application의 복구서비스 및 상황 감지를 위한 모니터링&알람 시스템
4. Multi-Cloud 시스템 구성

1. Private Cloud 구축(이번엔 멀티노드로!)
2. 클라우드 내 개발 환경/Application 배포시스템
3. Application의 복구서비스 및 상황 감지를 위한 모니터링&알람 시스템
4. ~~Multi-Cloud 시스템 구성~~(결국 이걸 건들지도 못했다...)



http://blog.daum.net/_blog/BlogTypeView.do?blogid=0BGdu&articleno=16869839&categoryId=941840®dt=20120626141404



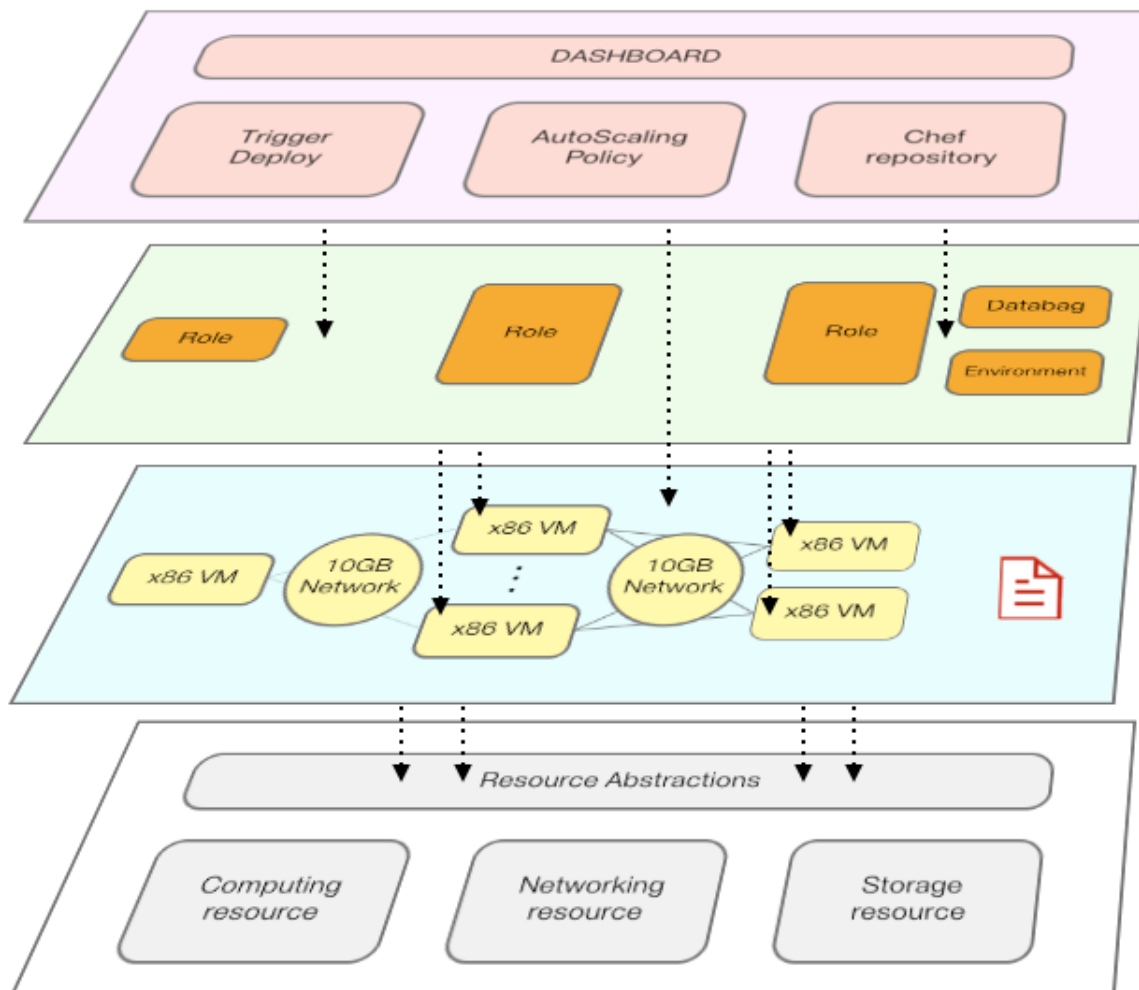
http://blog.daum.net/_blog/BlogTypeView.do?blogid=0BGdu&articleno=16869839&categoryId=941840®dt=20120626141404



<http://www.mdgfund.org/node/1957>



<http://theuranus.tistory.com/2195>



Dashboard

- AutoScaling Policy 정의
- Deploy Policy 정의

Deploy Automation

- application environment deploy
- application deploy

Monitoring & AutoScaling

- Scalable dedigned Infra Template
- Monitoring & Alerting

Cloud Infra

- support Infra API

그래서 나온 7개월의 결과물

Application Deploy - Open x

61.43.139.95/project/cdms_app_create/create/CDMS-3-tier-autoscaling.template/?display_type=app

CDMS Cloud Deploy

demo

admin 로그인

프로젝트

- Cloud Deploy
- Deploy List
- Infra Deploy
- Application Deploy
- Chef Management
- Compute
- 네트워크
- 관리자

Application Deploy

essential > ElasticLoadBalancer > WebServerGroup > Database

Single Instance Setting

Chef Role

- ✓ tomcat
- teletalkvi-tomcat

Instance auto-scaling policy

- MIN - vCPU: 1ea, RAM: 2GB, Disk: 10GB

1:3

- MAX - vCPU: 4ea, RAM: 8GB, Disk: 40GB

when

meter_name: cpu_util, period: 60, threshold: 50

then

UPPolicyScalingAdjustment: 1

when

meter_name: cpu_util, period: 60, threshold: 15

then

DOWNPoicyScalingAdjustment: -1

Template view

```
WebServerGroup:{
  "DependsOn": "WaitDBCondition",
  "Properties": {
    "AvailabilityZones": {
      "Fn::GetAZs": ""
    },
    "LaunchConfigurationName": {
      "Fn::Ref": "LaunchConfig"
    }
  }
}
```

Deploy Preview

ElasticLoadBalancer

ElasticLoadBalancer Instance

WebServerGroup

tomcat Instance, tomcat Instance, tomcat Instance, tomcat Instance

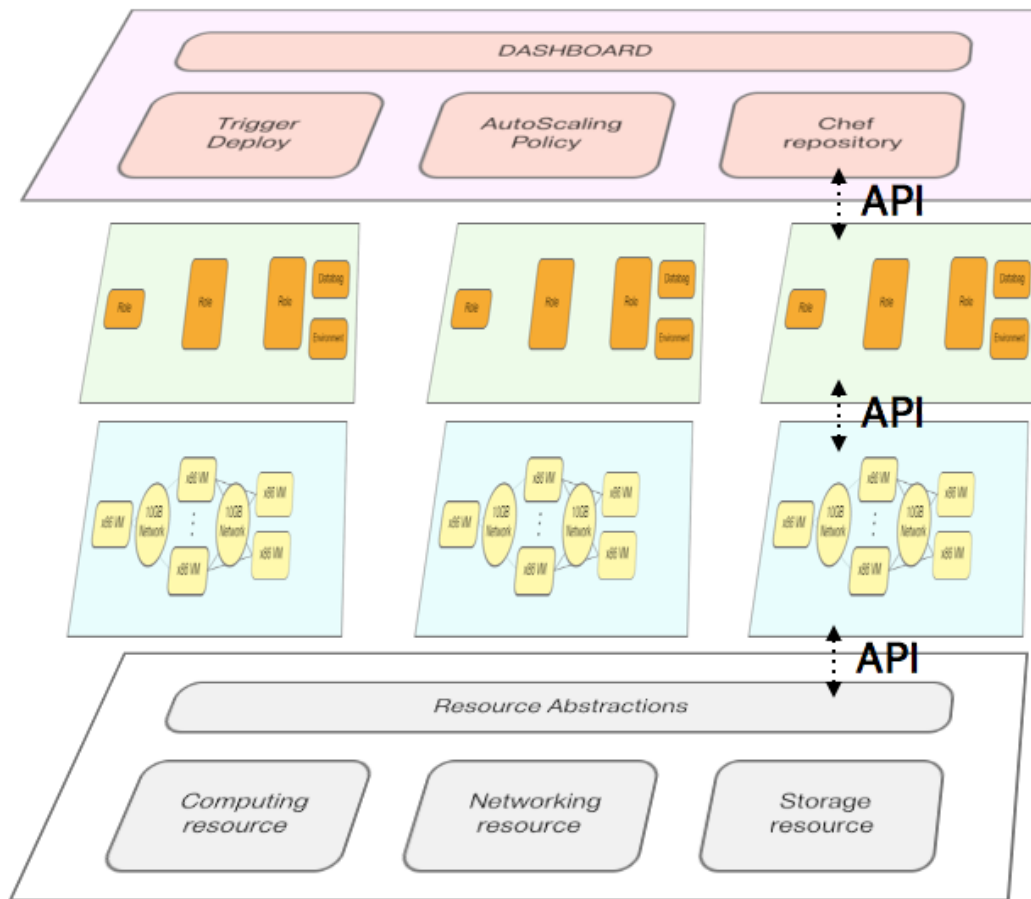
Database

mysql Instance

cancel next



<http://nbbcri.org/2014/02/17/break/>



Dashboard

- AutoScaling Policy 정의
- Deploy Policy 정의

Deploy Automation

- application environment deploy
- application deploy

Monitoring & AutoScaling

- Scalable dedigned Infra Template
- Monitoring & Alerting

Cloud Infra

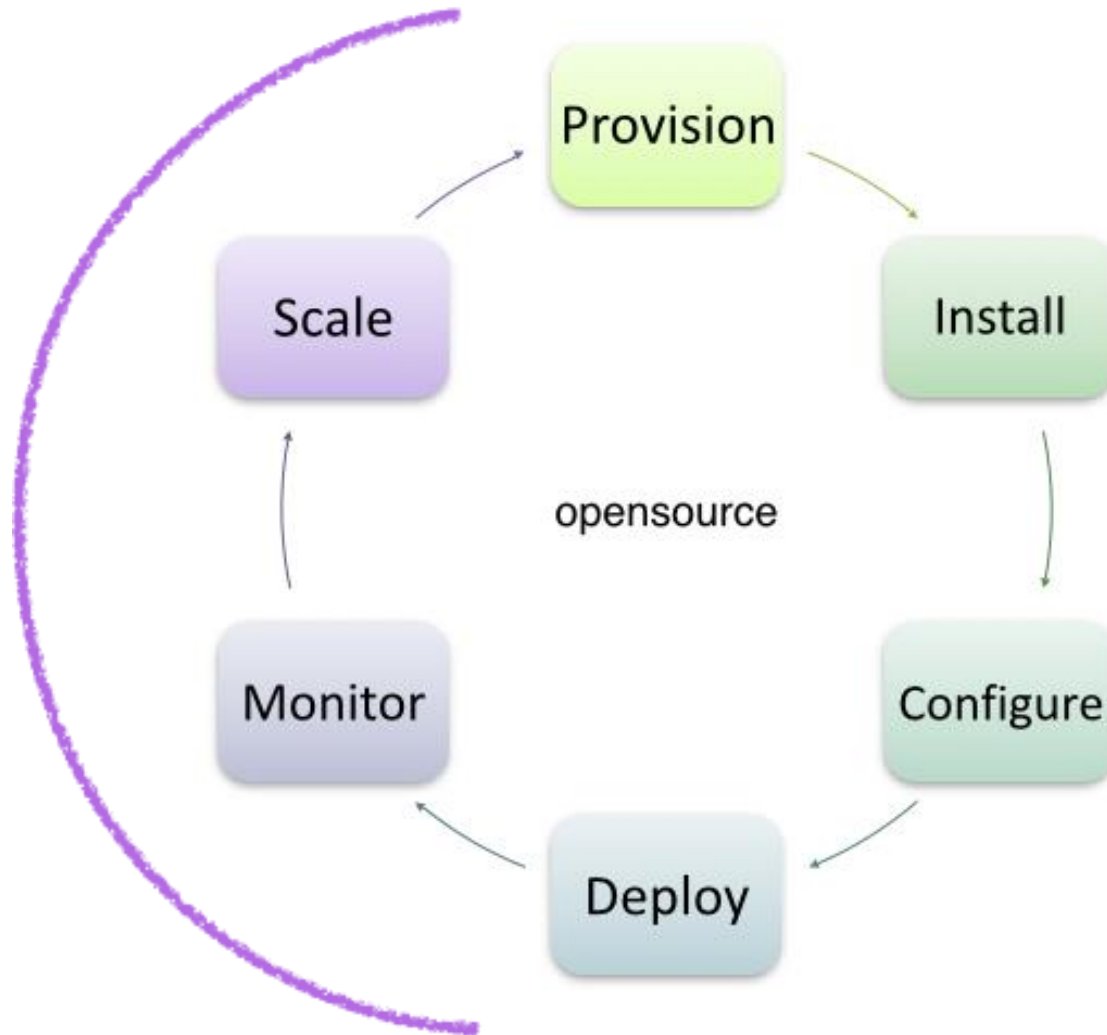
- support Infra API

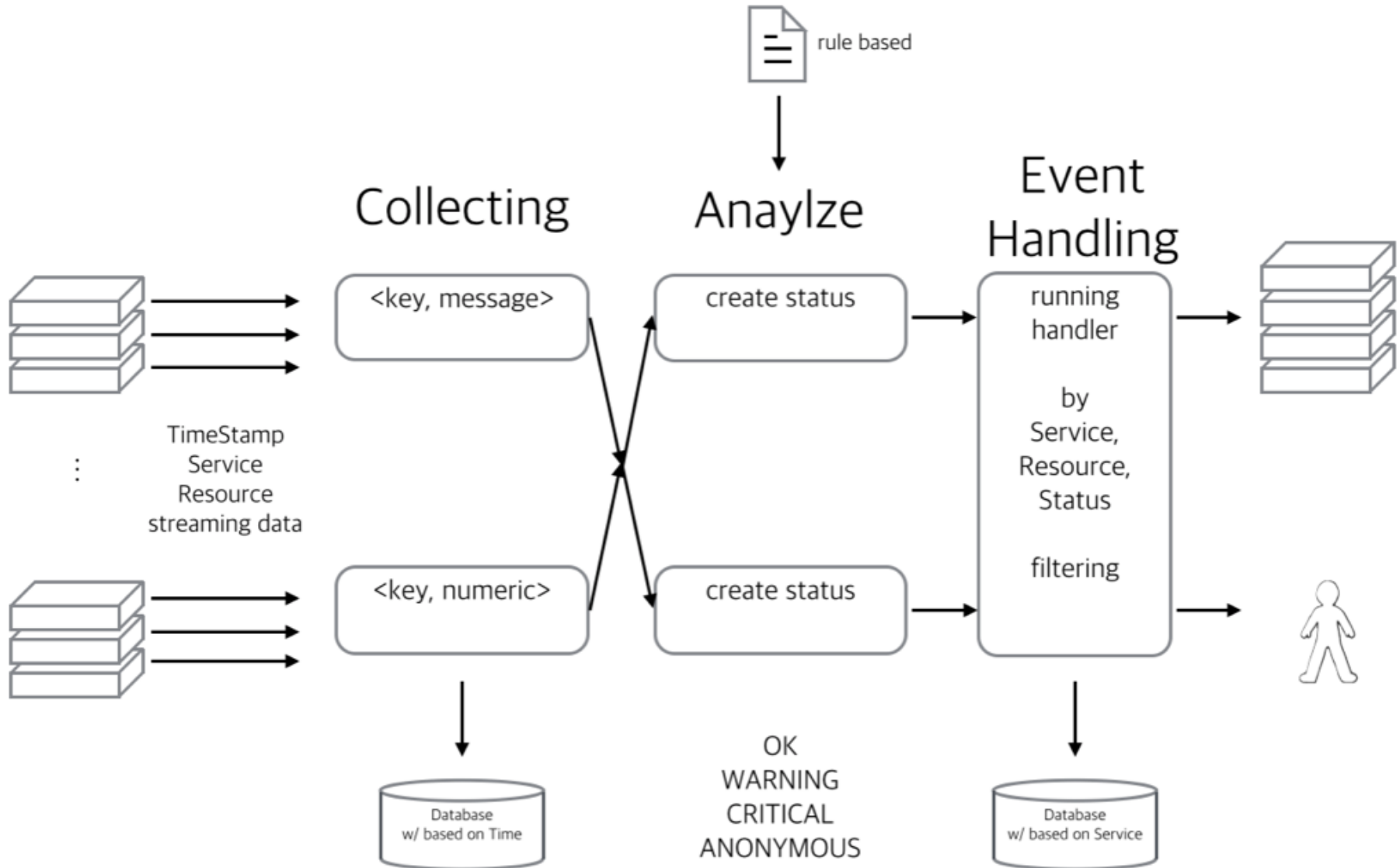


(진짜 진짜 감사합니다)



<http://salesbox.com/would-you-use-linkedin-if-it-only-had-your-friends-as-members/>



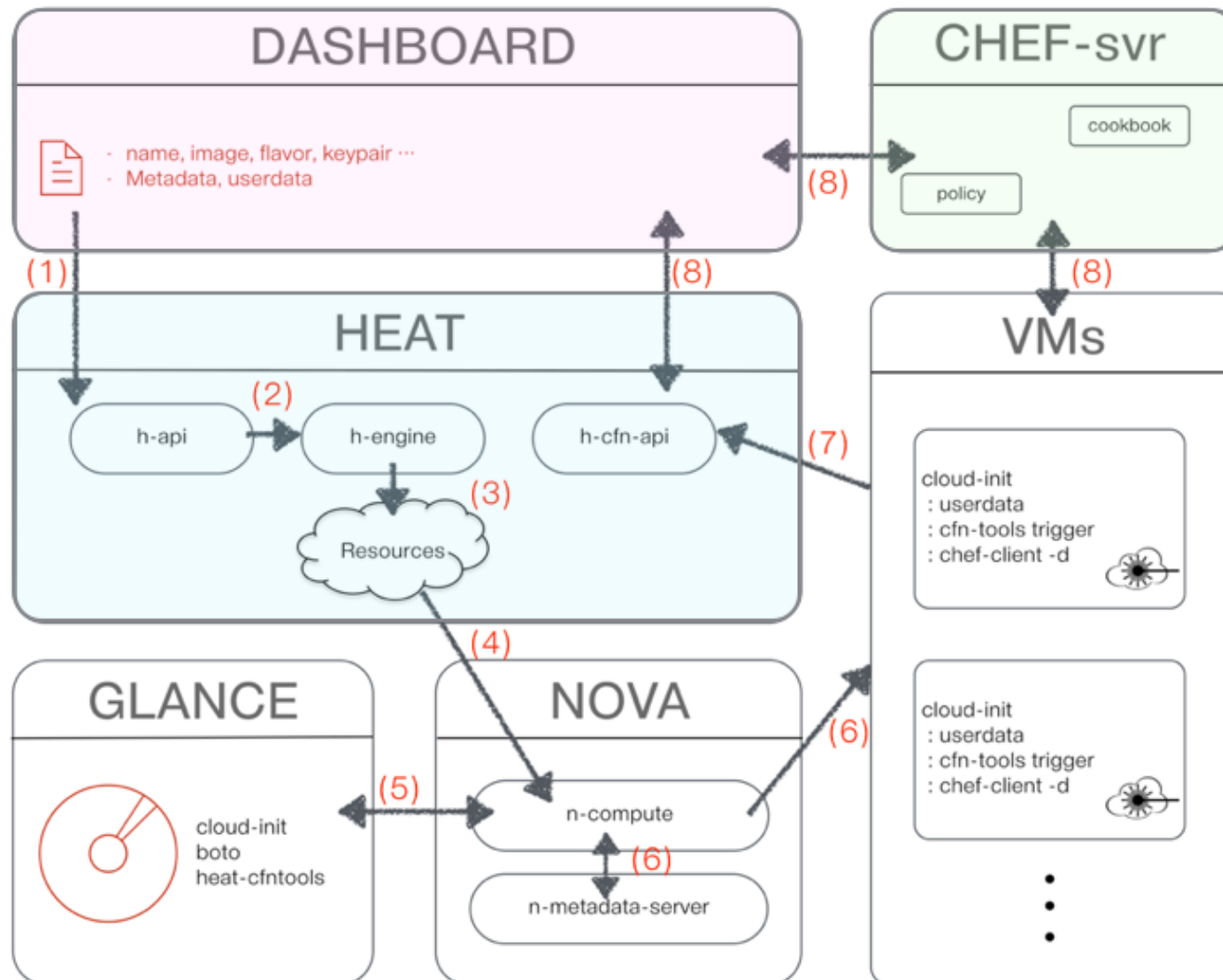




THANK YOU!

APPENDIX

CDMS Flow - initial provisioning



CDMS Flow - autoscaling

