

OPENSTACK DAYS
CHINA

Heat Optimization

Rico Lin

Chief OpenStack Technologist, inwinSTACK

IRC: ricolin

rico.l@inwinstack.com

@ricolintw





Orchestration

Why NOT > 80% adoption rates

Orchestrates multiple composite cloud applications by using either the native HOT template format or the AWS CloudFormation template format, through both an OpenStack-native REST API and a CloudFormation-compatible Query API.



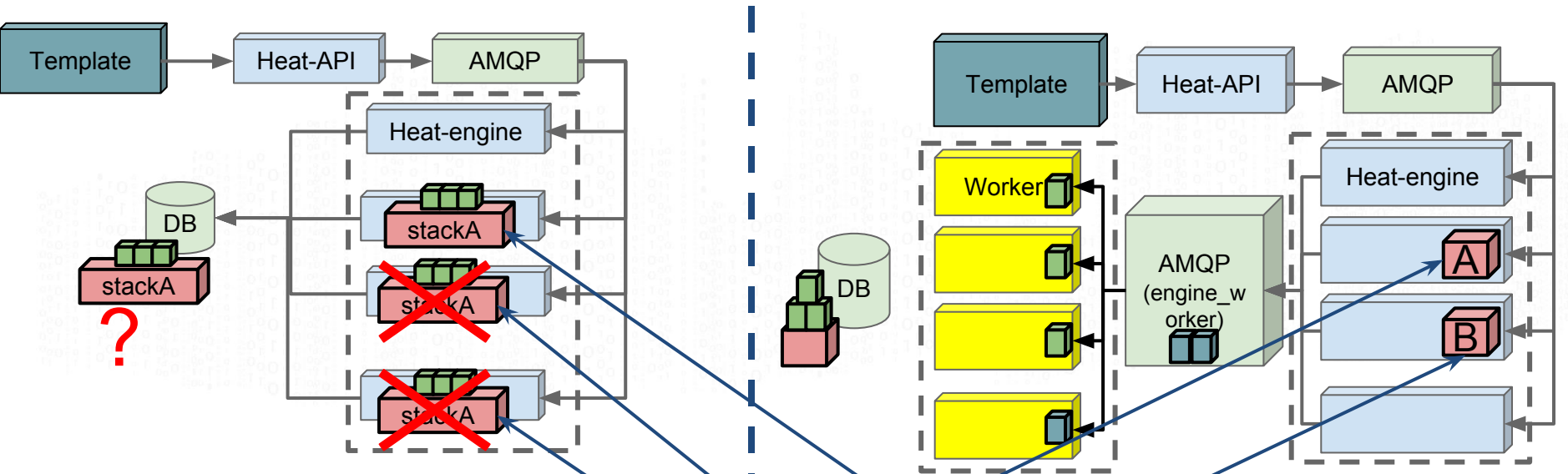
Because!

- Was not parallel execution
- Database monster
- Memory monster
- Don't know how we can use it?



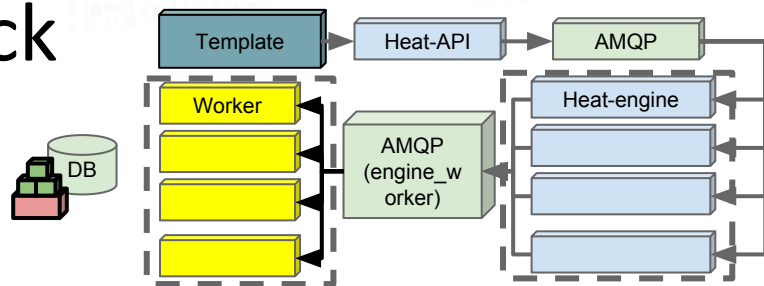
Convergence :

Truly using every power we have

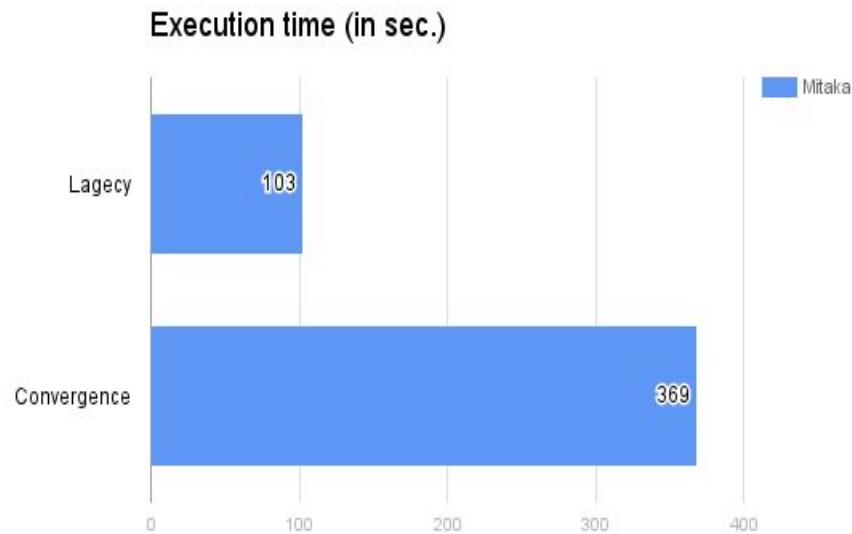
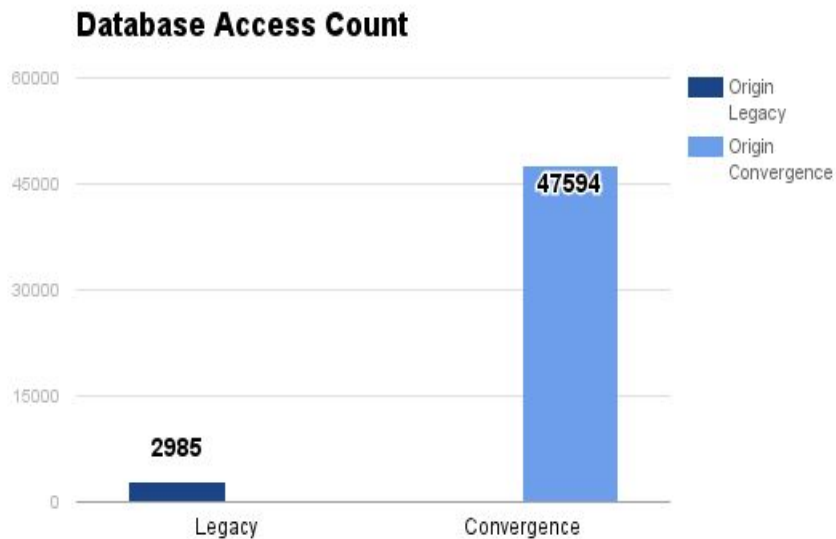


convergence_engine = True

- Start Newton, `convergence_engine` default to `True`
- `num_engine_workers = max(4, multiprocessing.cpu_count())`
- (WIP) Migrate your old stack to convergence_stack



More Database consume when we change to Convergence



Eager and Lazy Fetch

- What always been query right next to each other? (**Eager fetch it!**)
 - raw_template always after stack
- What we can took a chance and read if required (**Lazy fetch it!**)
 - stack_tag not always after stack

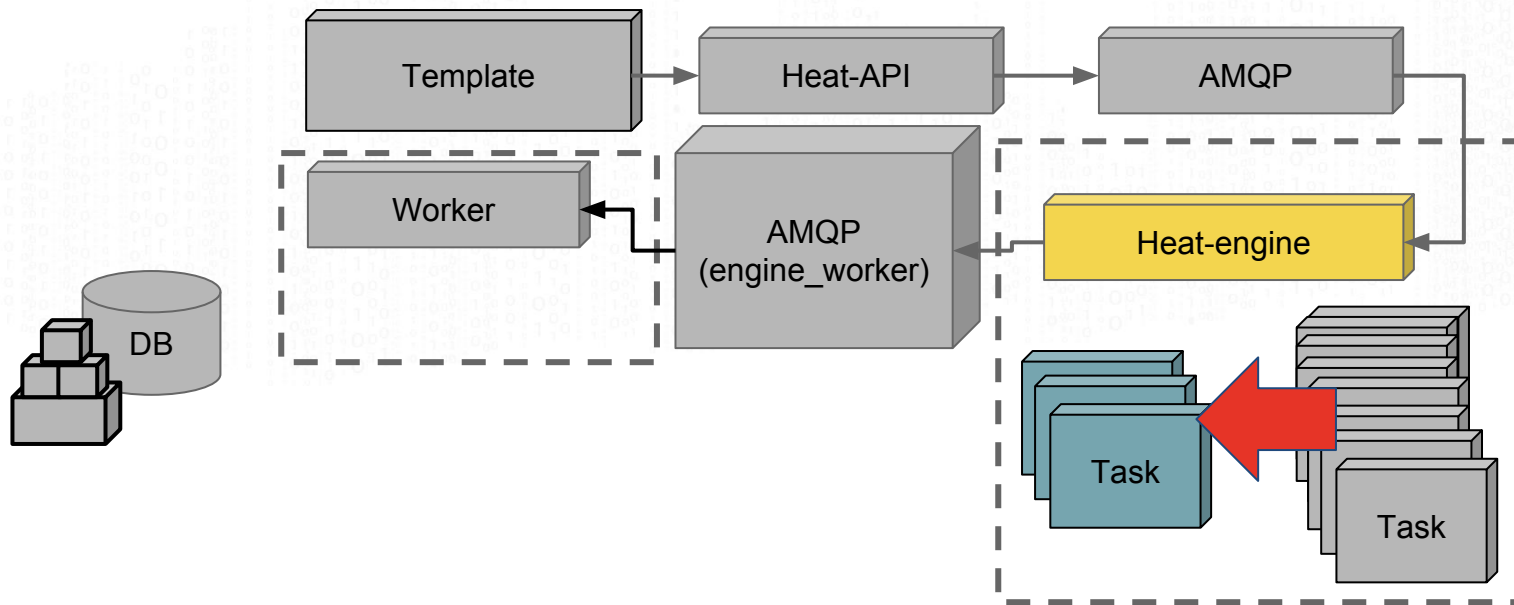
Contexts Cache

- Only apply when `refresh = False`
- Replace database access



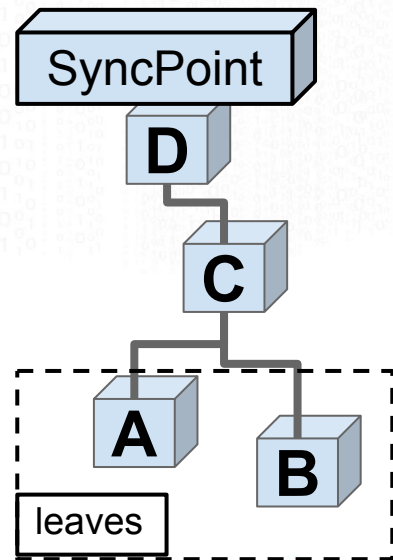
Batch Task

- Accept Queue to process tasks in batch



Optimize Sync Table

- Resolving back to back retry
 - Random retry
 - $\text{max_wait} = \text{conflict_count} * 0.01$
 - $\text{wait} = \text{random}(0, \text{max_wait})$
 - Exponential backoff
 - $\text{wait} = \text{random}(\text{max_wait}) * 2^N + \text{jitter}$



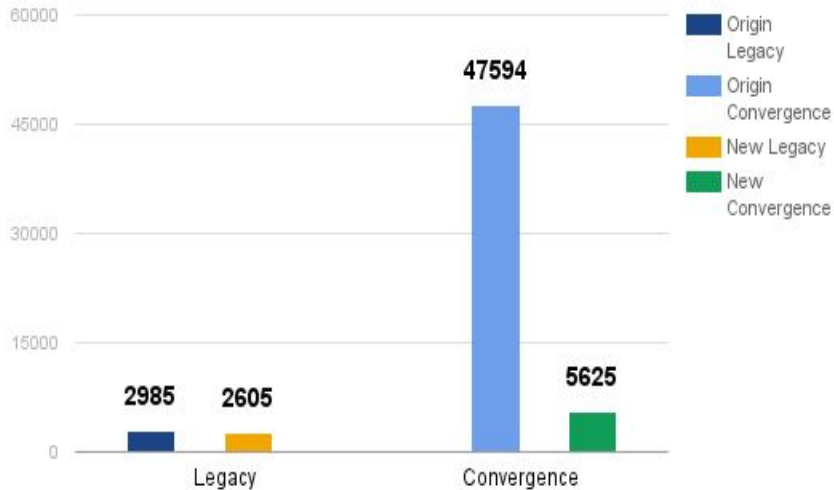
Resolving Update Cancel

- Under develop
- `handle_%s_cancel`

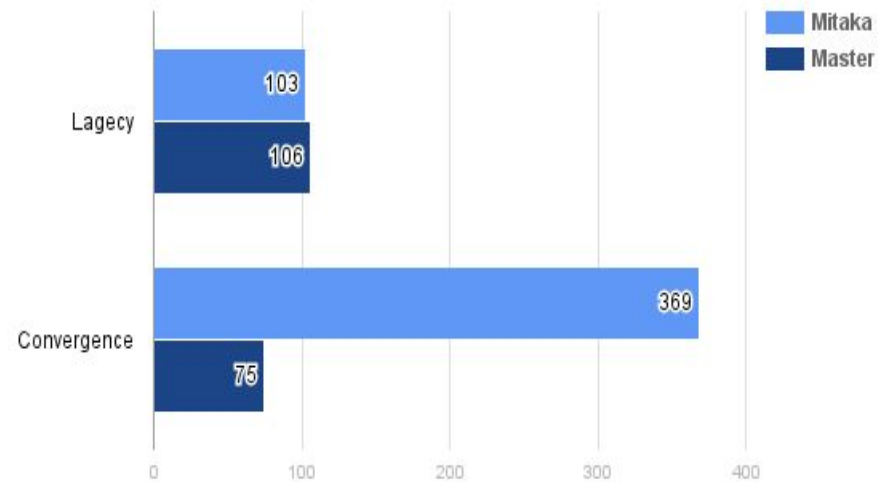


Reduce dramatically

Database Access Count

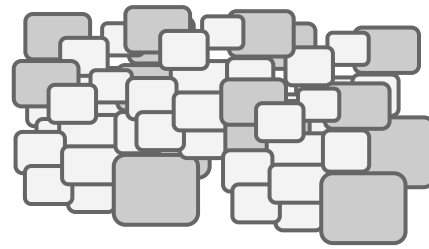


Execution time (in sec.)

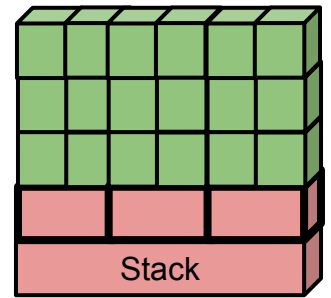


Optimize your script

More operate = More mistake



> 100 Operations
>100 Min.



VS
1 Operation
<20 Min.



Stack = Parameters + Resources

parameters:

flavor:

type: string

network:

type: string

resources:

service_server_group:

type: OS::Heat::ResourceGroup

properties:

count: { get_param: num_servers }

resource_def:

type: OS::Nova::Server

properties:

image: { get_param: image_id }

flavor: { get_param: flavor }

key_name: { get_param: key_name }

networks: [{ network: { get_param: network } }]

Select Template

Template Source *

File

File

Direct Input

URL

Description:

Use one of the available template source options to specify the template to be used in creating this stack.

Environment Source

File

Environment File ?

Choose File

No file chosen

Cancel

Next

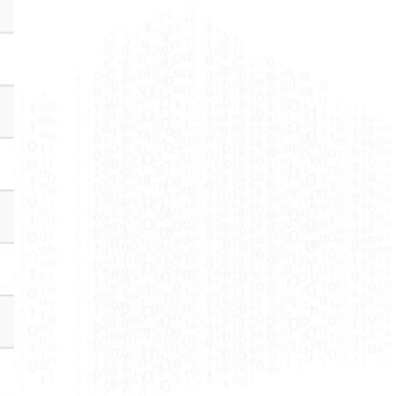


Functions

heat_template_version.2016-10-14

Function	Description
list_join	A function for joining one or more lists of strings.
yaql	A function for executing a yaql expression.
digest	A function for performing digest operations.
get_attr	A function for resolving resource attributes.
repeat	A function for iterating over a list of items.
resource_facade	A function for retrieving data in a parent provider template.
equals	A function for comparing whether two values are equal.
str_replace	A function for performing string substitutions.
get_resource	A function for resolving resource references.
map_merge	A function for merging maps.
str_split	A function for splitting delimited strings into a list.
get_param	A function for resolving parameter references.
get_file	A function for including a file inline.

Displaying 13 items



Easy Constraint, Easy life

Parameters:

flavor:

type: string

description: Flavor for the server to be created

constraints:

- **custom_constraint: nova.flavor**

volume_size:

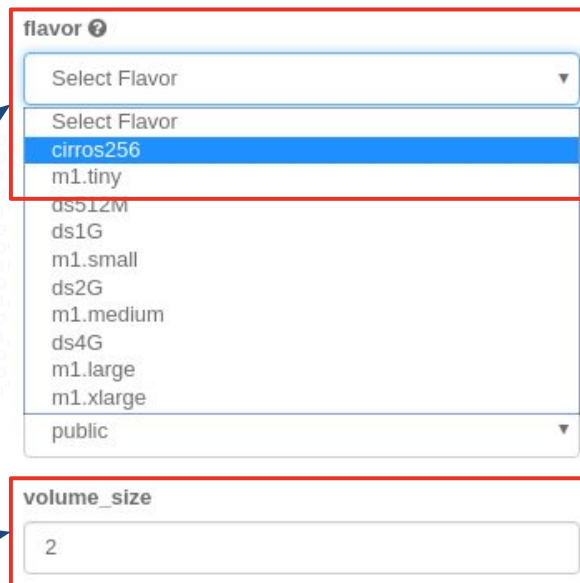
type: number

description: Size of volume to attach to instance

default: 2

constraints:

- **range: { min: 1, max: 10 }**



The image shows a screenshot of a web form with two fields highlighted by red boxes. The first field is a dropdown menu labeled 'flavor' with a help icon. The dropdown is open, showing a list of options: 'Select Flavor', 'cirros256' (highlighted in blue), 'm1.tiny', 'ds512M', 'ds1G', 'm1.small', 'ds2G', 'm1.medium', 'ds4G', 'm1.large', 'm1.xlarge', and 'public'. The second field is a text input labeled 'volume_size' containing the number '2'. Two blue arrows point from the text in the left column to these two fields.



A good way to monitor your resources

Topology Overview **Resources** Events Template

Stack Resource	Resource	Time Since Event	Status	Status Reason
asdf	f52e2857-9fff-4bf6-b924-038ce4828e05	20 minutes	Create Complete	Stack CREATE completed successfully
my_res_group	a7876416-effd-41db-8575-4358091ebdb2	20 minutes	Create Complete	state changed
my_res_group	-	20 minutes	Create In Progress	state changed
cinder_volume	e8606108-529b-419c-9442-c3164eb868ab	20 minutes	Create Complete	state changed
cinder_volume	-	20 minutes	Create In Progress	state changed
asdf	f52e2857-9fff-4bf6-b924-038ce4828e05	20 minutes	Create In Progress	Stack CREATE started

Displaying 6 items



Making Good Use of Old Resources

resources:

cinder_volume:

type: OS::Cinder::Volume

properties:

size: { get_param: volume_size }

image: { get_param: image }

service_server_group:

type: OS::Heat::ResourceGroup

properties:

count: { get_param: num_servers }

resource_def:

type: OS::Nova::Server

properties:

image: { get_param: image_id }

flavor: { get_param: flavor }

key_name: { get_param: key_name }

networks: [{ network: { get_param: network } }]

rgrs

Stack Resource	Resource	Stack Resource Type	Date Updated	Status	Actions
	my_res_group	OS::Heat::ResourceGroup	41 minutes	Create In Progress	state changed

➔ **Nothing change!**



Nested Stack -> type: stack.yaml

resources:

cinder_volume_attach:

type: cinder_volume_attach.yaml

properties:

size: { get_param: volume_size }

...

service_server_groups:

type: server_group.yaml

properties:

...

```
cinder_volume:
  type: OS::Cinder::Volume
  properties:
    size: { get_param: volume_size }
    image: { get_param: image }
```

```
service_server_group:
  type: OS::Heat::ResourceGroup
  properties:
    count: { get_param: num_servers }
    resource_def:
      type: OS::Nova::Server
      properties:
        image: { get_param: image_id }
        flavor: { get_param: flavor }
        key_name: { get_param: key_name }
        networks: [{network: {get_param: network}}]
```



Resource Group

```
my_res_group
  type: OS::Heat::ResourceGroup
  properties:
    count: 100
    resource_def:
      type: OS::Nova::Server
    ...
```

```
group:
  type: OS::Heat::AutoScalingGroup
  ...
  type: OS::Nova::Server
  ...
```

```
outputs:
  server_list:
    value: [{get_attr: [my_res_group, name]}]
```

```
scale_up_policy:
  type: OS::Heat::ScalingPolicy
  properties:
    auto_scaling_group_id: { get_resource: group }
  ...
```



Software Config

```
my_deps:
  type: OS::Heat::SoftwareDeploymentGroup
  properties:
    config: {get_resource: config}
    servers: {{get_attr: [servers, name]}}
    signal_transport: NO_SIGNAL
```

```
config:
  type: OS::Heat::SoftwareConfig
  properties:
    group: ansible
    inputs:
      - name: cluster-one
    outputs:
      - name: result
    config:
      get_file: config-scripts/cluster-ansible.ansible
```

```
---
- name: Mysql Cluster -
  connection: local
  hosts: localhost
  tasks:
    - name: Hello touch_file
      shell: echo {{ foo }} >> /tmp/{{ bar }}

    - name: Hello echo
      shell: echo "The file /tmp/{{ bar }} contains {{ foo }}"
    >> {{ heat_outputs_path }}.result
```



Multi layer services

```
...  
shell: ... >> {{ heat_outputs_path }}.result
```

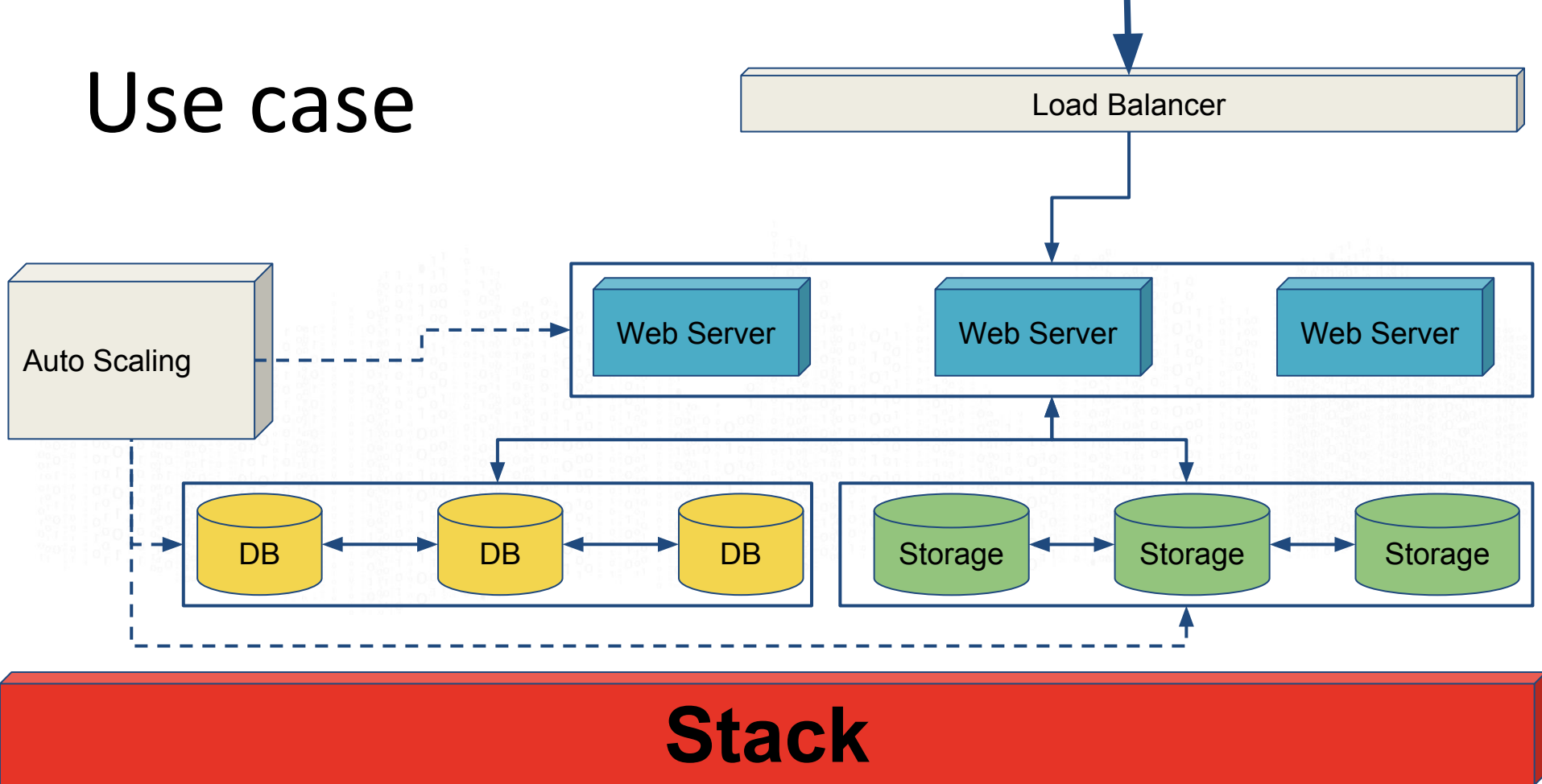
```
Another_config:  
  type: OS::Heat::SoftwareConfig  
  properties:  
    group: ansible  
  inputs:  
    - name: {get_attr: [config, result]}  
  outputs:  
    - name: result  
  config:  
    get_file: cluster_db-ansible.ansible
```

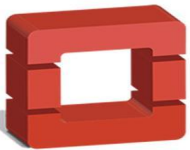
Another_SoftwareDeploymentGroup

```
config: {get_resource: another_config}  
servers: {{get_attr: [servers, name]}}  
signal_transport: NO_SIGNAL
```



Use case





OPENSTACK DAYS
CHINA

Heat Optimization

Demo

Rico Lin

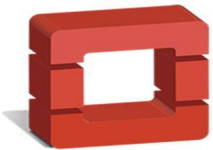
Chief OpenStack Technologist, inwinstack

IRC: ricolin

rico.l@inwinstack.com

@ricolintw





OPENSTACK DAYS
CHINA

Heat Optimization

Q&A

Rico Lin
Chief OpenStack Technologist, inwinstack

IRC: #openstack
@inwinstack
@ricolin

