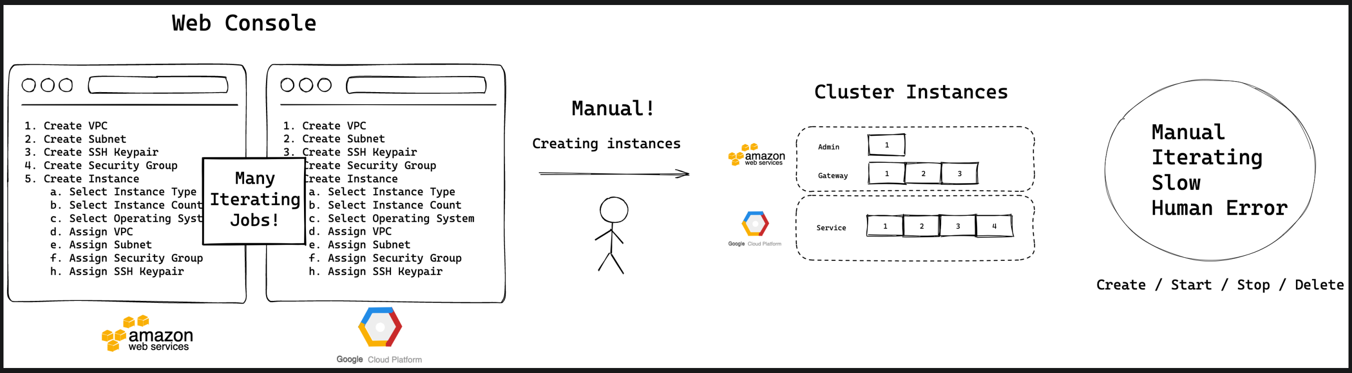
Cluster Management

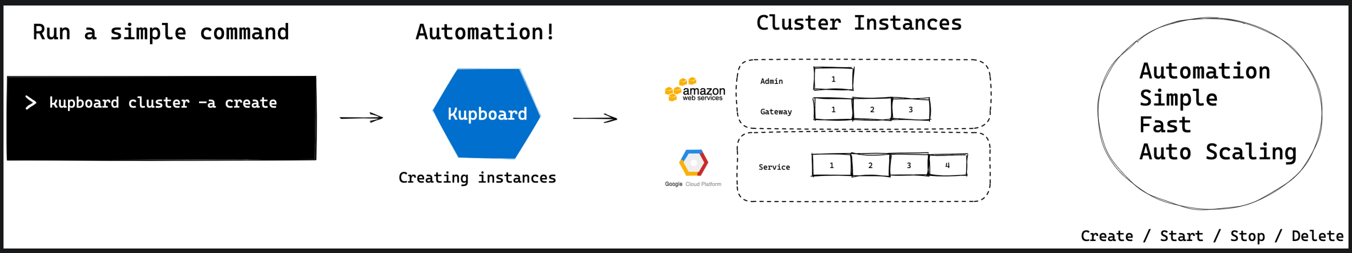
Cluster Management is the feature to manage server instancess on a cloud environment. Before `0.9.5`, the server instances must be prepared and the IP addresses of the servers need to be defined in `kupboard.yaml`. However, after `0.9.5`, server instances are automatically created by kupboard based on the cluster structure defined in `kupboard.yaml`.

You can also use commands to manage instances such as `create`, `delete`, `start`, and `stop`. These commands would be very useful when you want to add or remove a node, or change an instance type as different situations.

As shown below, Cluster Management automatically can handle many repetitive jobs that should be done manually.



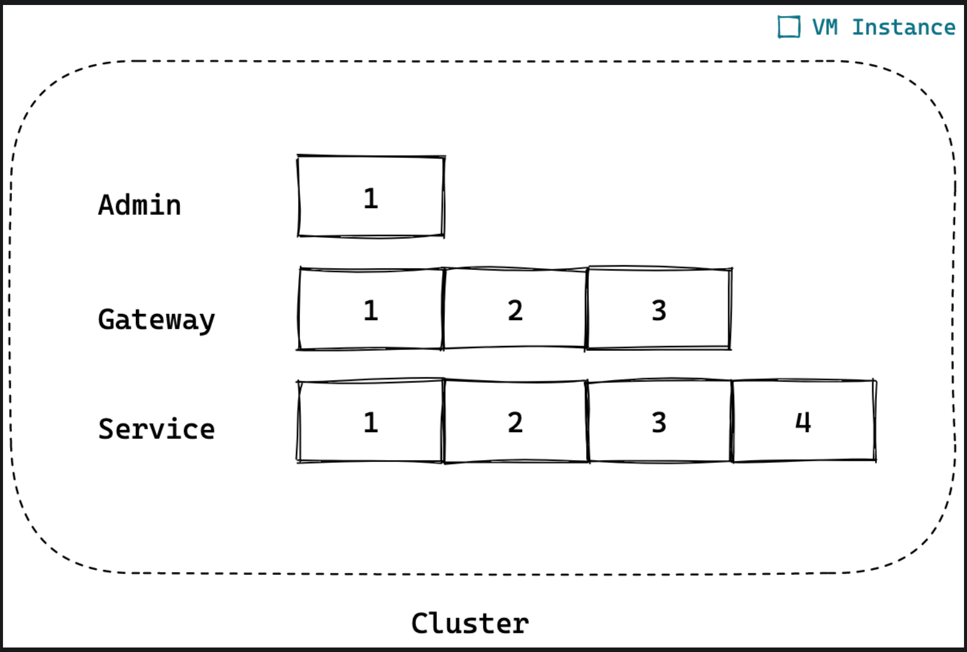
Manual Cluster Management



```
> kupboard cluster -a create
```

Automatic Cluster Management

Cluster



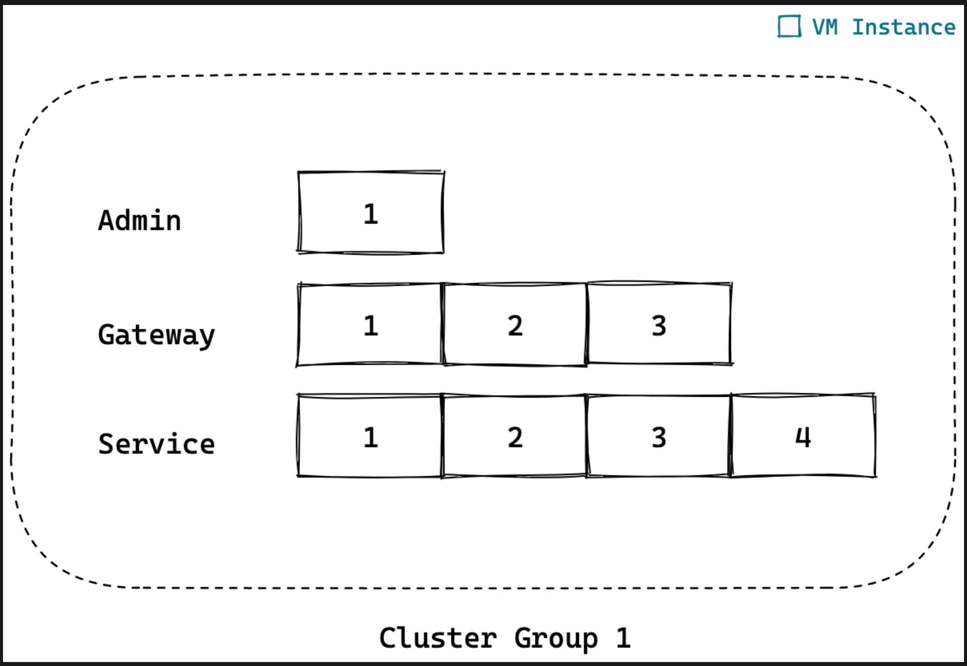
If you want to manage servers manually, you can build a service environment by defining the IP addresses of the servers in the cluster in `kupboard.yaml`. If the cluster contains 1 admin server, 3 gateway servers, and 4 service servers as shown above, the cluster definition would be as follows:

```
cluster:
  admin:
    - name: admin-node1
      public_ip: x.x.x.x
      private_ip: x.x.x.x

  gateway:
    - name: gateway-node1
      public_ip: x.x.x.x
      private_ip: x.x.x.x
    - name: gateway-node2
      public_ip: x.x.x.x
      private_ip: x.x.x.x
    - name: gateway-node3
      public_ip: x.x.x.x
      private_ip: x.x.x.x

  service:
    - name: service-node1
      public_ip: x.x.x.x
      private_ip: x.x.x.x
    - name: service-node2
      public_ip: x.x.x.x
      private_ip: x.x.x.x
    - name: service-node3
      public_ip: x.x.x.x
      private_ip: x.x.x.x
    - name: service-node4
      public_ip: x.x.x.x
      private_ip: x.x.x.x
```

Cluster Group



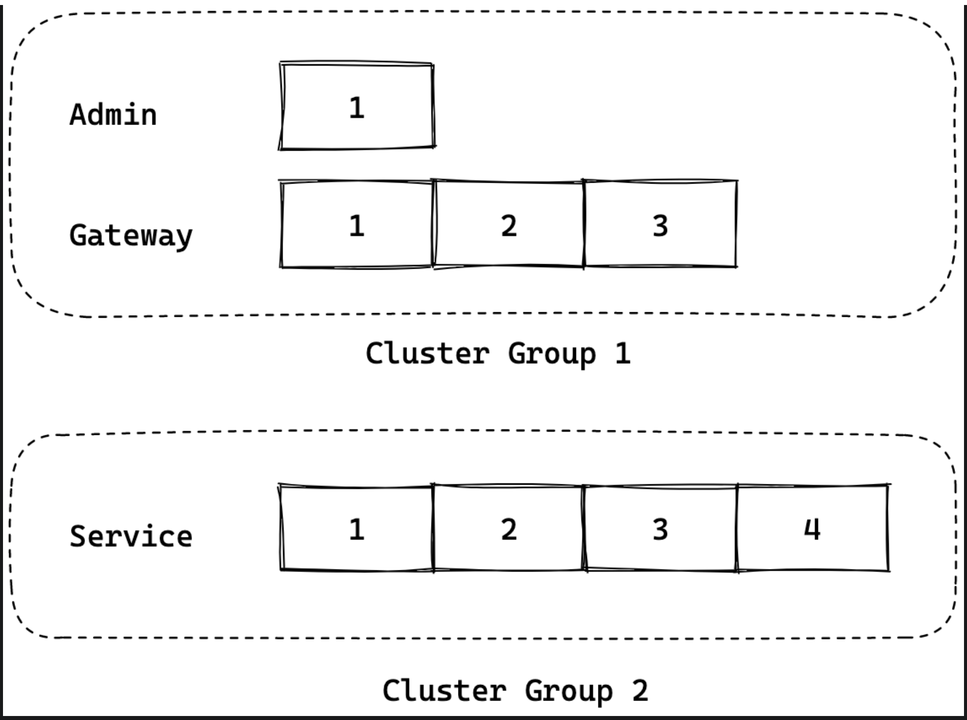
If you use Cluster Group, you only need to define the names of the clusters and the number of servers for clusters instead of all IP addresses of servers as follows. Based on this cluster structure, kupboard automatically creates server instances. For more detail, see Cluster Commands.

```
clusterGroups:
  - name: group1
    cluster:
      - name: admin
        count: 1
      - name: gateway
        count: 3
      - name: service
        count: 4
```

However, For kupboard to control server instances the credential and information of a cloud provider is required and they can be defined as below.

```
clusterGroups:
  - name: group1
    provider: aws
    credentials:
      access_key: "AWS_SECRET_KEY"
      secret_key: "AWS_ACCESS_KEY"
    instance:
      zone: ap-northeast-2a
      region: ap-northeast-2
      instance_type: "t3.large"
      ami: "ami-04876f29fd3a5e8ba" # ubuntu20.4
      volume_size: 50
      subnet_cidr: 172.31.0.0/20
    cluster:
      - name: admin
        count: 1
      - name: gateway
        count: 3
      - name: service
        count: 4
```

If you want to divide a cluster into multi groups, you can define groups as follows. This is related to Multi Cloud, please see Multi Cloud for more information



```
clusterGroups:
  - name: group1
    ...
    cluster:
      - name: admin
        count: 1
      - name: gateway
        count: 3
  - name: group2
    ...
    cluster:
      - name: service
        count: 4
```

Cluster Command

- Create or delete all cluster groups

```
$ kupboard cluster -a <create|delete>
```

- Create or delete a cluster group

```
$ kupboard cluster -a <create|delete> --group <group-name>
```

- Create, start, stop or delete an instance

```
$ kupboard cluster -a <create|start|stop|delete> --node <node-name>
```

- Update local cluster information from a cloud provider

```
$ kupboard cluster -a status
```

- Show current cluster information

```
$ kupboard cluster
```

Providers

The information required to manage instances depends on a cloud provider. Below are the examples for AWS, MS Azure and Google Cloud Platform.

> **NOTE**
>
> Currently Cluster Group supports AWS, MS Azure and Google Cloud Platform.

AWS

```
clusterGroups:
  - name: group1
    provider: aws
    credentials:
      access_key: "AWS_SECRET_KEY"
      secret_key: "AWS_ACCESS_KEY"
    instance:
      zone: ap-northeast-2a
      region: ap-northeast-2
      instance_type: "t3.large"
      ami: "ami-04876f29fd3a5e8ba" # ubuntu20.4
      volume_size: 50
      subnet_cidr: 172.31.0.0/20
    cluster:
      - name: admin
```

```
        count: 1
        instance_type: "t3.medium"
      - name: gateway
        count: 3
      - name: service
        count: 4
```

Google Cloud Platform

```
clusterGroups:
  - name: group2
    provider: gcp
    credentials:
      service_account_file: <gcp-key-file.json>
      project_id: <gcp-project-id>
    instance:
      instance_type: "n1-standard-1"
      source_image: projects/ubuntu-os-
cloud/global/images/family/ubuntu-1804-lts
      zone: asia-northeast3-a
      region: asia-northeast3
      volume_size: 50
      subnet_cidr: 10.240.0.0/24
    cluster:
      - name: admin
        count: 1
      - name: gateway
        count: 3
      - name: service
        count: 4
```

- `gcp-key-file.json` A key file of your google project and it must be located in `data/certs`.

Azure

```
clusterGroups:

  - name: group3
    provider: azure
    credentials:
      subscription_id: <subscription-id>
      client_id: <client-id>
      secret: <secret>
      tenant: <tenant-id>
    instance:
      instance_type: Standard_DS2_v2
      image_sku: 18.04-LTS # ubuntu
      region: koreacentral
      volume_size: 50
      subnet_cidr: 10.240.0.0/24
    cluster:
      - name: admin
        count: 1
      - name: gateway
        count: 3
      - name: service
        count: 4
```

Setup

When you create a cluster by using the `cluster` command, the default user with a root permission is automatically created. The username is `kupboard` for GCP and Azure and, `ubuntu` for AWS.

```
$ kupboard setup --init-user --root-username <username>
```

Edit this page