Logging Handbook

NEAR ME

Elasticsearch

Deploying Elasticsearch

Elasticsearch is running on a single EC2 instance within the VPC. OpsWorks is used to deploy Elasticsearch.

Chef Cookbook: https://github.com/mdyd-dev/chef-recipes/tree/nearme/elastic

OpsWorks (us-west-1)

- Stack: nm-production
- Layer: Elasticsearch Logs

Use OpsWorks Deployments to install or update:

1. update_custom_cookbooks
2. execute_recipes (elastic::default)

Snapshots

To be able to backup and restore the Elasticsearch instance, Elasticsearch snapshots are used. The S3 Repository plugin allows to store snapshots on S3 (installed via Chef).

A Lambda function is used to trigger an Elasticsearch snapshot daily.

Another Lambda function is cleaning up old Elasticsearch snapshots with a retention period of 7 days.

Git Repository: https://bitbucket.org/widdix/elasticsearch-snapshot (Needs to be transferred to NEAR ME).

Restoring a snapshot

Create a snapshot repository pointing to the S3 bucket where the Elasticsearch snapshots are stored.

```
# replace <ES_ENDPOINT>, <BUCKET_NAME> and <REGION>
curl -H 'Content-Type: application/json' -X PUT -d
'{"type":"s3","settings":{"bucket":"<BUCKET_NAME>","region":"<REGI
ON>"}}' <ES_ENDPOINT>/_snapshot/s3restore
```

```
# example
curl -H 'Content-Type: application/json' -X PUT -d
'{"type":"s3","settings":{"bucket":"elasticsearch-snapshot-snapsho
tbucket-16mrg59yalfpt","region":"us-west-1"}}'
http://localhost:9200/_snapshot/s3restore
```

List available snapshots.

```
# replace <ES_ENDPOINT>
curl <ES_ENDPOINT>/_snapshot/s3restore/_all

# example
curl http://localhost:9200/_snapshot/s3restore/_all
```

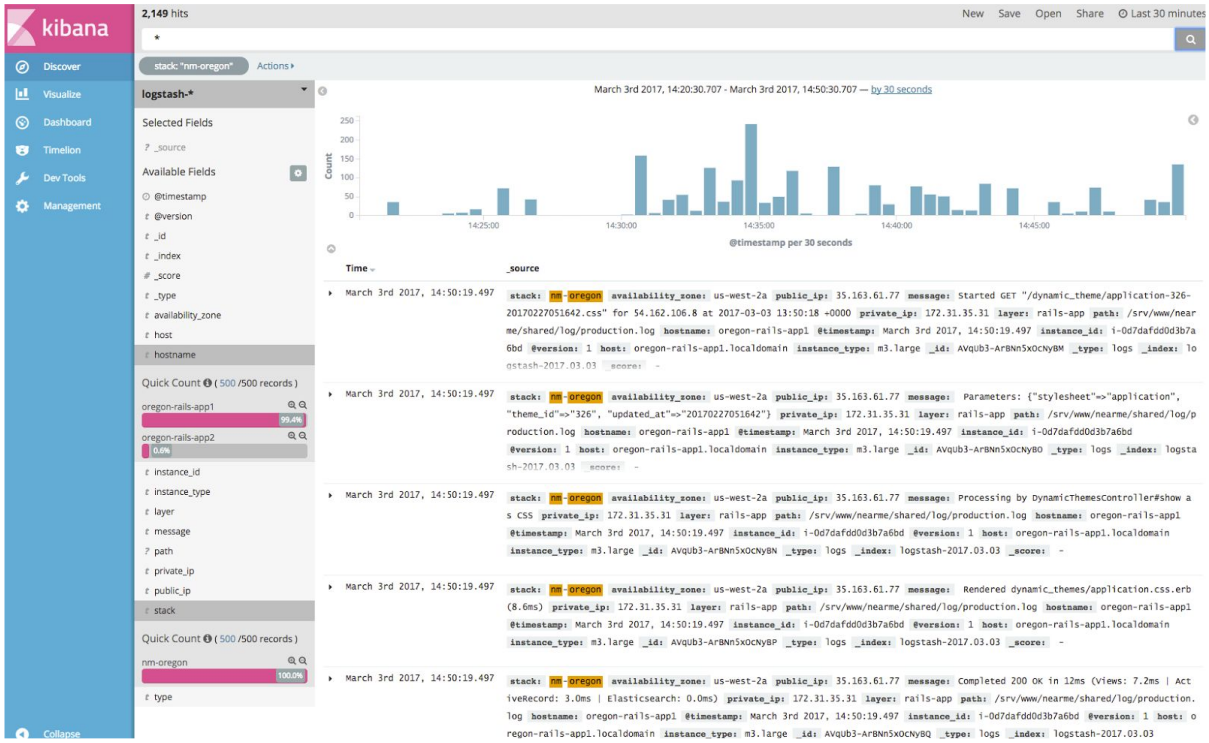
Restore snapshot including all indices named logstash-*.

```
# replace <ES_ENDPOINT>, <INDICES>, and <SNAPSHOT>
curl -H 'Content-Type: application/json' -X POST -d '{"indices":
"<INDICES>"}'
<ES_ENDPOINT>/_snapshot/s3restore/<SNAPSHOT>/_restore

# example
curl -H 'Content-Type: application/json' -X POST -d '{"indices":
"logstash-*,elb-*"}'
http://localhost:9200/_snapshot/s3restore/1488456887147.snapshot/_
restore
```

Verify state of restore.

```
# replace <ES_ENDPOINT> and <SNAPSHOT>
curl <ES_ENDPOINT>/_snapshot/s3restore/<SNAPSHOT>/_status

#example
curl
http://localhost:9200/_snapshot/s3restore/1488456887147.snapshot/_
status
```

Kibana



Accessing Kibana

Elasticsearch and Kibana are only accessible within VPC.

Use a SSH tunnel to access Kibana from your local machine.

Add the following configuration to your `~/.ssh/config` file.

```
Host nearme-us-west-1
  Hostname 52.9.77.133
  User <USERNAME>
  LocalForward 5601 172.31.26.214:5601

Host nearme-us-west-2
  Hostname 35.164.192.225
  User <USERNAME>
  LocalForward 5601 172.31.38.166:5601
```

Open a SSH connection.

```
ssh nearme-us-west-1
```

```
# or
ssh nearme-us-west-2
```

Point your browser to http://localhost:5601/app/kibana

Discovering logs

Kibana shows two different index types:

- logstash-* is containing logs from EC2 instances and applications
- elb-* is containing logs from ELBs

EC2 Logs

Logstash is used to forward logs (e.g. application logs and system logs) from the EC2 instance to Elasticsearch. OpsWorks is used to deploy Logstash on application instances.

Logstash is deployed to the following OpsWorks stacks: nm-staging and to nm-production.

Chef Cookbook: https://github.com/mdyd-dev/chef-recipes/tree/nearme/logstash

Use OpsWorks Deployments to install or update:

3. update_custom_cookbooks
4. execute_recipes (logstash::default)

ELB Logs

An AWS Lambda function is to forward ELB logs to Elasticsearch. Whenever a new log file is created within the S3 bucket, the Lambda function is triggered. The function is then downloading and parsing the log file. Afterwards the log messages are put into Elasticsearch.

Git Repository: https://bitbucket.org/widdix/elb-logs-to-elasticsearch (Needs to be transferred to NEAR ME).

Infrastructure Changes

2017-02-10

- Created a VPC endpoint for S3 in us-west-1.
- Launched an EC2 instance for Elasticsearch/Kibana (t2.medium with 100 GB EBS volume).
- Deployed Logstash to nm-production and nm-staging (Rails App instances).
- Deployed a Lambda function which is forwarding ELB logs to Elasticsearch.

    - Lambda function
    - Security Group for the Lambda function referenced by the Security Group attached to Elasticsearch
    - S3 events configuration to near-me-production-elb-logs

2017-02-20

- Deployed a Lambda function which is triggering Elasticsearch snapshots
    - S3 bucket created to store snapshots in
    - Security Group for the Lambda function referenced by the Security Group attached to Elasticsearch
    - Added IAM policy to IAM role aws-opsworks-ec2-role allowing read/write access to S3 bucket storing Elasticsearch snapshots

2017-03-03

- Deployed a Lambda function which is cleaning up Elasticsearch snapshots
- Allowed incoming TCP traffic on port 9200 from bastion host to Elasticsearch (Logs) to be able to use the API from an engineer's machine
- Logging infrastructure in us-west-2
    - Launched an EC2 instance for Elasticsearch/Kibana (t2.medium with 100 GB EBS volume).
    - Added Security Group nm-oregon-es-logs
    - Added Elasticsearch Logs layer to nm-oregon stack (OpsWorks)
    - Deployed logstash agent to the rails app layer within nm-staging-oregon and nm-oregon
    - Setup Elasticsearch snapshots (Lambda, S3 Bucket, Security Group)
    - Forwarding logs from ELB to Elasticsearch
        - Lambda function
        - Security Group for the Lambda function referenced by the Security Group attached to Elasticsearch
        - Added S3 events configuration to near-me-oregon-elb-logs
        - Added VPC Endpoint for S3