Inside the Elastic Stack

Testing & releasing a well known Open Source stack

Alexander Reelsen
@spinscale

Agenda

One slide every 30 seconds is ok, right?

1. Introduction
2. Elasticsearch testing & X-Pack integration
3. CI
4. Preparing & testing a release
5. Release

Agenda

One slide every 30 seconds is ok, right?

1. Introduction
2. Elasticsearch testing & X-Pack integration
3. CI
4. Preparing & testing a release
5. Release

Introduction

- Myself
  - Based in Munich, joined Elastic in early 2013
  - Developer, Elasticsearch & X-Pack Alerting
  - Organizes Search Meetup Munich

- Elastic
  - distributed, more than 550 employees across more than 30 countries
  - few offices & home office
  - Offering support subscriptions, training, consulting, EaaS & on premise

What is the Elastic Stack?

elastic

What is the Elastic Stack?

Welcome to the Zoo!

- Elasticsearch: Java
- Logstash: JRuby + Java
- Beats: Go
- Kibana: Javascript on node.js
- Commercial: C++, Scala, all of the above
- Tooling: Perl, python, node
- Clients: Ruby, PHP, Perl, python, groovy

It’s complicated

es	Nov 5, 2014 **1.4**	May 23, 2015 **1.5**	Jun 9, 2015 **1.6**	Jul 16, 2015 **1.7**
kibana	Feb 19, 2015 **4.0**		Jun 10, 2015 **4.1**	
ls	May 14th, 2015 **1.5**			
beats	May 27th, 2015 **1.0 Beta 1**	July 13th, 2015 **1.0 Beta 2**	Sept 4 th, 2015 **1.0 Beta 3**	


5.0
5.0.1
5.0.2
5.1.1
5.1.2
5.2.0
5.2.1
5.2.2

Agenda

One slide every 30 seconds is ok, right?

1. Introduction
2. Elasticsearch testing & X-Pack integration
3. CI
4. Preparing & testing a release
5. Release

Tests everywhere!

- Unit Tests
- REST Tests
- Packaging Tests
- Documentation Tests
- BWC tests
- Performance Tests

Unit Tests

```
public class JarHellTests extends ESTestCase {

 public void testRequiredJDKVersionTooOld() throws Exception
     Path dir = createTempDir();
     List<Integer> current = JavaVersion.current().getVersion
     List<Integer> target = new ArrayList<>(current.size());
```

Unit Tests

```
/**
 * Base testcase for randomized unit testing with Elasticsearch
 */
@Listeners({
        ReproduceInfoPrinter.class,
        LoggingListener.class
})
@ThreadLeakScope(Scope.SUITE)
@ThreadLeakLingering(linger = 5000) // 5 sec lingering
@TimeoutSuite(millis = 20 * TimeUnits.MINUTE)
@LuceneTestCase.SuppressSysoutChecks(bugUrl = "we log a lot on purpose")
// we suppress pretty much all the lucene codecs for now, except asserting
// assertingcodec is the winner for a codec here: it finds bugs and gives clear exceptions.
@SuppressCodecs({
        "SimpleText", "Memory", "CheapBastard", "Direct", "Compressing", "FST50", "FSTOrd50",
        "TestBloomFilteredLucenePostings", "MockRandom", "BlockTreeOrds", "LuceneFixedGap",
        "LuceneVarGapFixedInterval", "LuceneVarGapDocFreqInterval", "Lucene50"
})
@LuceneTestCase.SuppressReproduceLine
public abstract class ESTestCase extends LuceneTestCase {
```

Unit Tests

```
/**
 * Base testcase for randomized unit testing with Elasticsearch
 */
@Listeners({
        ReproduceInfoPrinter.class,
        LoggingListener.class
})
@ThreadLeakScope(Scope.SUITE)
@ThreadLeakLingering(linger = 5000) // 5 sec lingering
```

Integration Tests

```
/**
 * A test that keep a singleton node started for all tests that can be used to get
 * references to Guice injectors in unit tests.
 */
public abstract class ESSingleNodeTestCase extends ESTestCase {
```

```
/**
 * {@link ESIntegTestCase} is an abstract base class to run integration
 * tests against a JVM private Elasticsearch Cluster. The test class supports 2 different
 * cluster scopes.
 */
public abstract class ESIntegTestCase extends ESTestCase {
```

REST tests

```
{
  "nodes.info": {
    "documentation": "http://www.elastic.co/guide/en/elasticsearch/reference/master/cluster-nodes-info.html",
    "methods": ["GET"],
    "url": {
      "path": "/_nodes",
      "paths": ["/_nodes", "/_nodes/{node_id}", "/_nodes/{metric}", "/_nodes/{node_id}/{metric}"],
      "parts": {
        "node_id": {
          "type": "list",
          "description": "A comma-separated list of node IDs or names to limit the returned information; use `_local`
        },
        "metric": {
          "type": "list",
          "options": ["settings", "os", "process", "jvm", "thread_pool", "transport", "http", "plugins", "ingest"],
          "description": "A comma-separated list of metrics you wish returned. Leave empty to return all."
        }
      },
```

REST tests

```
---
"node_info test":
  - do:
      cluster.state: {}

  # Get master node id
  - set: { master_node: master }

  - do:
      nodes.info: {}

  - is_true: nodes
  - is_true: cluster_name
  - is_true: nodes.$master.roles
```

Packaging Tests

```
~/devel/elasticsearch/elasticsearch  master  rg 'config.vm.box ' Vagrantfile
26:    config.vm.box = "elastic/ubuntu-14.04-x86_64"
30:    config.vm.box = "elastic/ubuntu-16.04-x86_64"
40:    config.vm.box = "elastic/debian-8-x86_64"
44:    config.vm.box = "elastic/debian-9-x86_64"
48:    config.vm.box = "elastic/centos-6-x86_64"
52:    config.vm.box = "elastic/centos-7-x86_64"
56:    config.vm.box = "elastic/oraclelinux-6-x86_64"
60:    config.vm.box = "elastic/oraclelinux-7-x86_64"
64:    config.vm.box = "elastic/fedora-25-x86_64"
68:    config.vm.box = "elastic/opensuse-42-x86_64"
72:    config.vm.box = "elastic/sles-12-x86_64"
```

```
~/devel/elasticsearch/elasticsearch  master  ls -l qa/vagrant/src/test/resources/packaging/tests
total 152
-rw-r--r--  1 alr  staff   4671 Aug 29 09:49 20_tar_package.bats
lrwxr-xr-x  1 alr  staff     33 Dec  2  2016 25_tar_plugins.bats -> module_and_plugin_test_cases.bash
-rw-r--r--  1 alr  staff   6579 Aug  2 11:38 30_deb_package.bats
-rw-r--r--  1 alr  staff   6366 Aug  2 11:38 40_rpm_package.bats
lrwxr-xr-x  1 alr  staff     33 Dec  2  2016 50_modules_and_plugins.bats -> module_and_plugin_test_cases.bash
-rw-r--r--  1 alr  staff   8744 Aug 29 09:49 60_systemd.bats
-rw-r--r--  1 alr  staff   5492 Aug 29 09:49 70_sysv_initd.bats
-rw-r--r--  1 alr  staff   3729 Aug  2 11:38 80_upgrade.bats
-rw-r--r--  1 alr  staff   2544 Jul 13 09:16 90_reinstall.bats
-rw-r--r--  1 alr  staff  14661 Aug 29 09:49 module_and_plugin_test_cases.bash
```

Packaging Tests via BATS

```
@test "[DEB] install package" {
    dpkg -i elasticsearch-$(cat version).deb
}

@test "[DEB] package is installed" {
    dpkg -s 'elasticsearch'
}

@test "[DEB] verify package installation" {
    verify_package_installation
}

@test "[DEB] verify elasticsearch-plugin list runs without any plugins installed" {
    local plugins_list=`$ESHOME/bin/elasticsearch-plugin list`
    [[ -z $plugins_list ]]
}

@test "[DEB] elasticsearch isn't started by package install" {
    # Wait a second to give Elasticsearch a change to start if it is going to.
    # This isn't perfect by any means but its something.
    sleep 1
    ! ps aux | grep elasticsearch | grep java
    # You might be tempted to use jps instead of the above but that'd have to
    # look like:
    # ! sudo -u elasticsearch jps | grep -i elasticsearch
    # which isn't really easier to read than the above.
}

@test "[DEB] test elasticsearch" {
    start_elasticsearch_service
    run_elasticsearch_tests
}
```

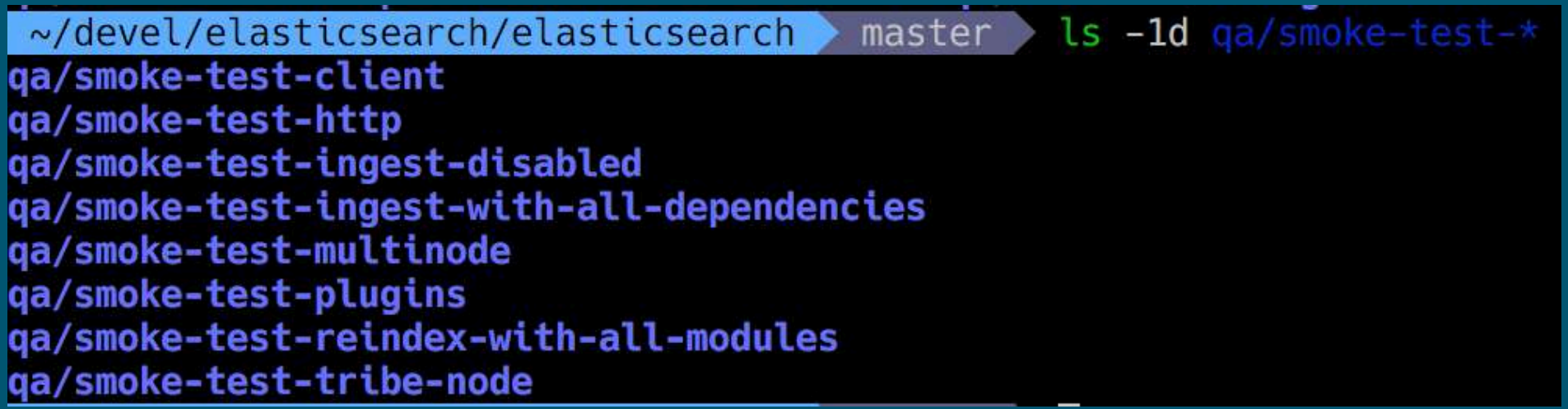
Documentation tests

```
~/devel/elasticsearch/elasticsearch master rg -N -B 4 -A 2 'CONSOLE' docs/reference/search/search.asciidoc
[source,js]
--------------------------------------------------
GET /twitter/_search?q=user:kimchy
--------------------------------------------------
// CONSOLE
// TEST[setup:twitter]
```

Documentation tests

```
[source,js]
----------------------------------------------------
PUT twitter/tweet/_bulk?refresh
{"index":{"_id":1}}
{"user" : "kimchy", "post_date" : "2009-11-15T14:12:12", "message" : "trying out Elasticsearch"}
{"index":{"_id":2}}
{"user" : "kimchi", "post_date" : "2009-11-15T14:12:13", "message" : "My username is similar to @kimchy!"}
----------------------------------------------------
// CONSOLE
// TESTSETUP

When sent a valid query:

[source,js]
----------------------------------------------------
GET twitter/_validate/query?q=user:foo
----------------------------------------------------
// CONSOLE

The response contains `valid:true`:

[source,js]
----------------------------------------------------
{"valid":true,"_shards":{"total":1,"successful":1,"failed":0}}
----------------------------------------------------
// TESTRESPONSE
```

Smoke tests



BWC tests

```
~/devel/elasticsearch/elasticsearch master ls -1d qa/*cluster* qa/r*
qa/full-cluster-restart
qa/mixed-cluster
qa/multi-cluster-search
qa/reindex-from-old
qa/rolling-upgrade
```

QA tests

```
~/devel/elasticsearch/elasticsearch master ls -1 qa
auto-create-index
evil-tests
full-cluster-restart
mixed-cluster
multi-cluster-search
no-bootstrap-tests
query-builder-bwc
reindex-from-old
rolling-upgrade
smoke-test-client
smoke-test-http
smoke-test-ingest-disabled
smoke-test-ingest-with-all-dependencies
smoke-test-multinode
smoke-test-plugins
smoke-test-reindex-with-all-modules
smoke-test-tribe-node
vagrant
verify-version-constants
wildfly
```

X-Pack

```
~/devel/elasticsearch ls -1d elasticsearch elasticsearch-extra/x-pack-elasticsearch
elasticsearch
elasticsearch-extra/x-pack-elasticsearch
~/devel/elasticsearch
```

Agenda

One slide every 30 seconds is ok, right?

1. Introduction
2. Elasticsearch testing & X-Pack integration
3. CI
4. Preparing & testing a release
5. Release

Jenkins

search

log in

Jenkins

ENABLE AUTO REFRESH

- People
- Build History
- Project Relationship
- Check File Fingerprint

Build Queue (5)

- elastic / elasticsearch # master - intake
- elastic / elasticsearch # 6.x - intake
- elastic / docs # master - build
- jenkins / git - fetch reference repos » fedora
- elastic / elasticsearch # master - macrobenchmark - periodic

Build Executor Status

master

1 Idle
2 Idle
3 Idle
4 Idle
5 Idle
6 Idle
7 elastic / elasticsearch # master #1885
8 elastic / elasticsearch # 5.6 #183
9 Idle

All | Docs | Elasticsearch | Lucene | **Main**

S	W	Name ↓	Last Success	Last Failure	Last Duration
		apache / lucene # branch_6x	25 min - #17187	1 hr 34 min - #17185	20 min
		apache / lucene # master	47 min - #17345	6 hr 9 min - #17334	21 min
		apache / lucene # master - nfs	7 mo 1 day - #51	4 mo 10 days - #259	1 hr 8 min
		elastic / docs # master - build	55 min - #6323	1 day 6 hr - #6270	29 min
		elastic / elasticsearch # 2.4	1 mo 14 days - #127	14 days - #128	1 hr 56 min
		elastic / elasticsearch # 2.4 - intake	14 days - #10	N/A	46 min
		elastic / elasticsearch # 5.0 - docs check	17 days - #4	N/A	5 min 29 sec
		elastic / elasticsearch # 5.1 - docs check	17 days - #4	N/A	5 min 27 sec
		elastic / elasticsearch # 5.2 - docs check	17 days - #5	N/A	5 min 12 sec
		elastic / elasticsearch # 5.3 - docker alpine	9 hr 44 min - #23	17 days - #22	1 hr 25 min
		elastic / elasticsearch # 5.3 - docs check	9 hr 44 min - #14	28 days - #11	4 min 57 sec
		elastic / elasticsearch # 5.4 - docs check	7 days 8 hr - #17	28 days - #13	5 min 35 sec
		elastic / elasticsearch # 5.5	15 days - #105	7 hr 13 min - #150	2 hr 55 min
		elastic / elasticsearch # 5.5 - aggressive opts	8 hr 16 min - #72	N/A	1 hr 5 min
		elastic / elasticsearch # 5.5 - backward compatibility tests	3 hr 56 min - #151	18 days - #95	1 hr 31 min

Jenkins

search

Jenkins ▸ elastic / elasticsearch # master - unix compatibility ▸

- Back to Dashboard
- Status
- Changes
- Workspace
- GitHub

Build History trend

find x

#1543	Sep 22, 2017 8:23 PM
#1542	Sep 22, 2017 12:23 PM
#1541	Sep 22, 2017 4:23 AM
#1540	Sep 21, 2017 8:23 PM
#1539	Sep 21, 2017 12:23 PM
#1538	Sep 21, 2017 4:23 AM
#1537	Sep 20, 2017 8:23 PM

Project elastic / elasticsearch # master - unix compatibility

Project name: elastic+elasticsearch+master+multijob-unix-compatibility
The Elasticsearch master branch Unix compatibility tests.

Configurations

amazon centos debian fedora opensuse oraclelinux sles ubuntu

Latest Test Result (no failures)

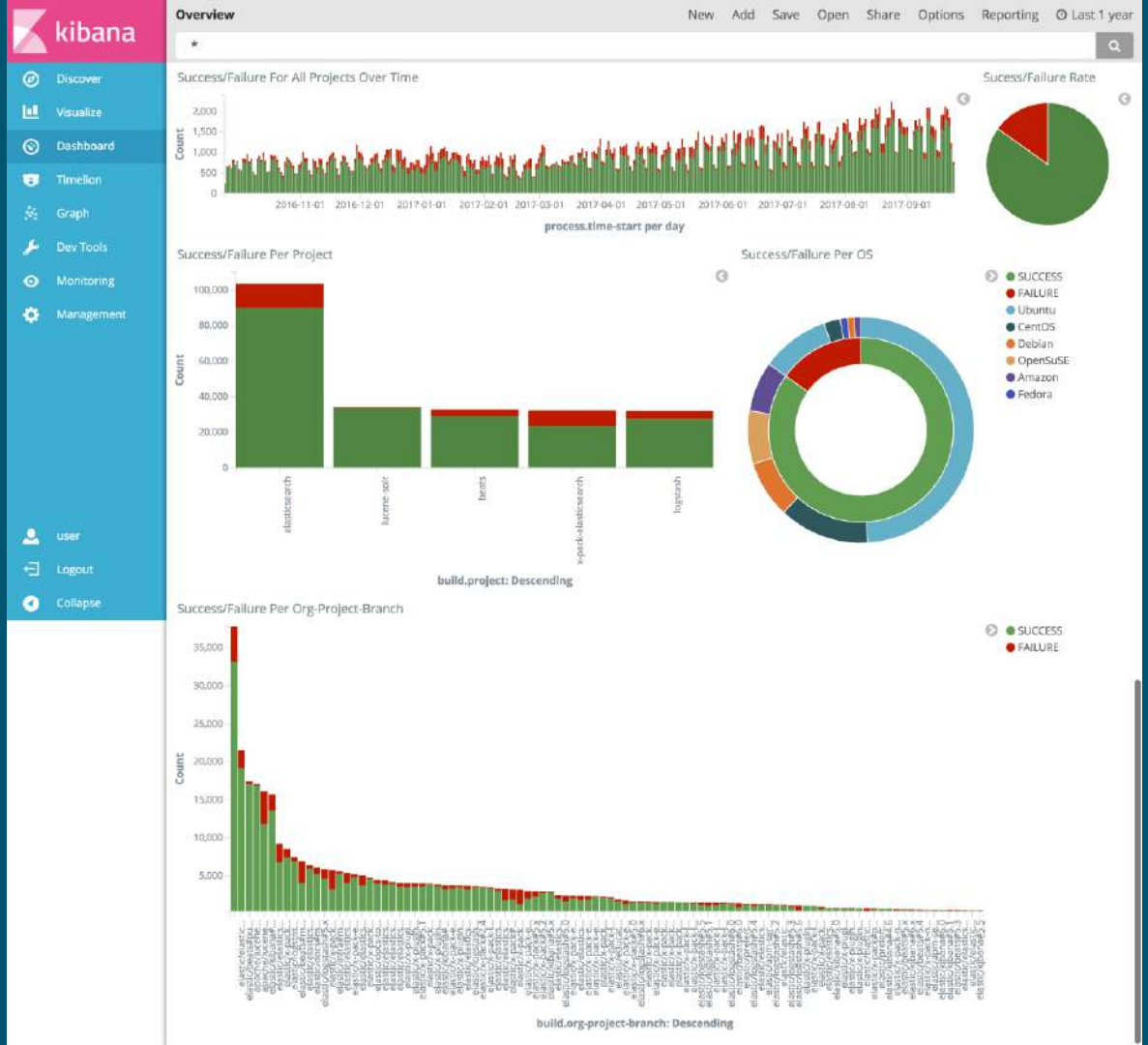
Permalinks

- Last build (#1543), 1 hr 42 min ago
- Last stable build (#1543), 1 hr 42 min ago
- Last successful build (#1543), 1 hr 42 min ago
- Last failed build (#1542), 9 hr 41 min ago
- Last unsuccessful build (#1542), 9 hr 41 min ago
- Last completed build (#1543), 1 hr 42 min ago

Upstream Projects

elastic / elasticsearch # master

elastic


kibana
Overview
New
Add
Save
Open
Share
Options
Reporting
Last 1 year
Discover
Visualize
Dashboard
Timelion
Graph
Dev Tools
Monitoring
Management
user
Logout
Collapse
Success/Failure For All Projects Over Time
Count
process.time-start per day
Sucess/Failure Rate
Success/Failure Per Project
Count
elasticsearch
lucene-solr
beats
x-pack-elasticsearch
logstash
build.project: Descending
Success/Failure Per OS
SUCCESS
FAILURE
Ubuntu
CentOS
Debian
OpenSuSE
Amazon
Fedora
Success/Failure Per Org-Project-Branch
Count
build.org-project-branch: Descending

Elastic Benchmarks

This page lists the various benchmarks we run for different projects within Elastic. They are created by the developers themselves, and focus on the most important performance metrics.

Elasticsearch

We run benchmarks oriented on spotting performance regressions in metrics such as indexing throughput or garbage collection times.

Geonames

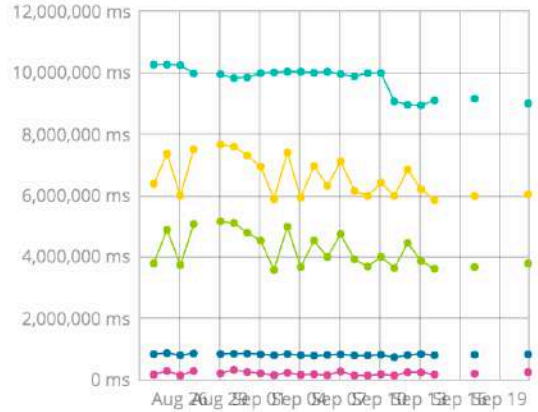
Indexing 11 million location documents and running various full text queries (match, function_score, ...) and aggregations.

Geopoint

Indexing 60 million location documents and running various geo-based queries (polygon, bounding box, distance, ...).

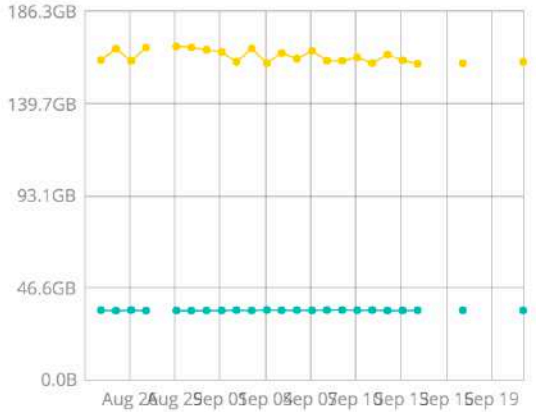
Aug 26 Aug 29 Sep 01 Sep 04 Sep 07 Sep 10 Sep 13 Sep 16 Sep 19

nightly-logging-index_times



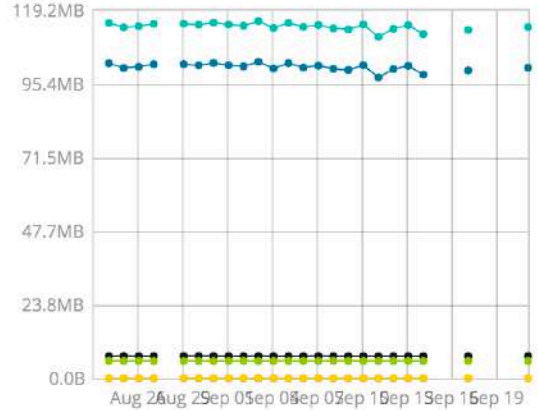
Indexing Time	9,008,179 ms
Merge time	6,041,489 ms
Refresh time	823,811 ms
Flush time	242,596 ms
Merge throttle ti	3,789,120 ms

nightly-logging-io



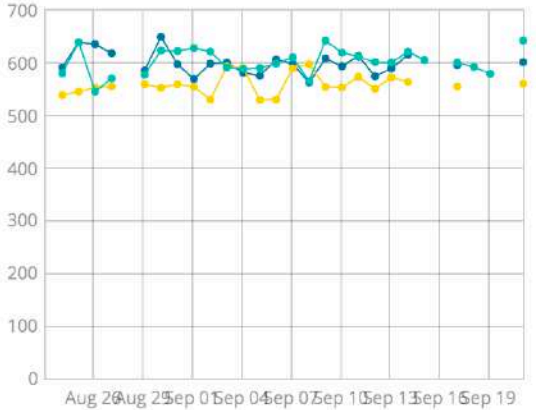
Index Size	35.0GB
Bytes Written	160.8GB

nightly-logging-segment_memory



Segments	113.9MB
Doc Values	106.9KB
Terms	100.7MB
Norms	35.1KB
Points	5.7MB
Stored Fields	7.3MB

nightly-logging-segment_count



Append (Defaults)	643
Append (4gheap)	561
Append Sorted (4gheap)	602

Moar testing

- 750+ builds ES + x-pack per day across all branches
- Testing of pull requests
- Testing on different operating systems
- Testing on different JVMs
- Testing with different garbage collectors
- Testing of feature branches
- Dedicated jenkins cluster for x-pack as well as other tools
- Docs CI: Ensure cross reference links work
- Test triage, developers take care of build failures

Agenda

One slide every 30 seconds is ok, right?

1. Introduction
2. Elasticsearch testing & X-Pack integration
3. CI
4. Preparing & testing a release
5. Release

elastic

This repository Search Pull requests Issues Marketplace Explore

Code Issues 210 Pull requests 2 Projects 0 Wiki Settings Insights

5.5.0 #774

Edit New issue

Closed **clintongormley** opened this issue on 6 Jun · 85 comments

Timeline

Latest Unified Release Builds

Releasing - One project to build them all!

- Idea: Every engineer should be able to build a release candidate

- Single repo with `Vagrantfile` and `gradle` build
- Requirements: Vagrant (+ winrm), VirtualBox, gradle
- Vagrant image installs the technology zoo!
- Builds projects, creates package repositories (+ signing) and uploads to S3 bucket, uploads to sonatype staging repo

- Result: Final immutable artifacts, which are ready to be released

Agenda

One slide every 30 seconds is ok, right?

1. Introduction
2. Elasticsearch testing & X-Pack integration
3. CI
4. Preparing & testing a release
5. Release

Releasing

```
gradle release
```

Releasing

- Uploading/building documentation
- Publish blog posts
- Update website download links
- Forum posts

- Post release work (internal version updates in ES)

Elastic Product End of Life Dates

We love all our products, but sometimes we must say goodbye to a release so that we can continue moving forward on future development and innovation. Our End of Life policy defines how long a given release is considered supported, as well as how long a release is considered still in active development or maintenance. We provide more information about **support policy**, **platform support**, and **support SLAs** separately.

Summary

The upshot is that we support each major release of our products for 18 months from the General Availability date, and we actively maintain the last minor release of the two most recent major branches of Elasticsearch, and compatible releases of Kibana, Beats, and Logstash. The rest of this document describes this philosophy in more details and provides concrete examples. Tables at the bottom of this page detail maintenance schedule for each of our supported products.

Types of Releases

Major versions, such as 1.0.0, 2.0.0, 5.0.0, and 6.0.0 provide us with an opportunity to introduce features and break backwards compatibility. Minor versions, such as 1.1.0 and 1.2.0, provide us with an opportunity to introduce features. Maintenance releases, such as 1.1.1 and 1.1.2, fix bugs only. Maintenance activity occurs on all releases, but we focus on the minor release stream (e.g. 1.1.x) to define how long we maintain a particular code line. Active maintenance of a minor release implies that we are fixing bugs and backporting some number of fixes into that code branch.

Summary

- Automate your releases!
- Do it as early as possible!
- Ensure everyone can do a release!
- Non technical aspects are incredibly harder to automate!

- Does this work for you? No idea!

Thanks for listening!

Questions?

@spinscale
alex@elastic.co

One more thing...

Elastic{ON} 2018

The Official Elasticsearch User Conference
February 26 - March 1, San Francisco

Call for Presentations
Open through October 31

Cause Awards Applications
Open through December 15

Links

- https://benchmarks.elastic.co
- https://elasticsearch-ci.elastic.co/
- https://beats-ci.elastic.co/
- https://kibana-ci.elastic.co/
- https://logstash-ci.elastic.co/
- https://github.com/elastic/rally
- https://speakerdeck.com/elasticsearch/oscon-scaling-a-distributed-engineering-team-from-50-250

Thanks for listening!

Questions?

@spinscale
alex@elastic.co