

Ein(Blick) in die Maschinerie einer
Continuous Delivery Pipeline:
Automatisierte Testaggregation und
-auswertung für angehende DevOps

@dataduke

@dastianoro

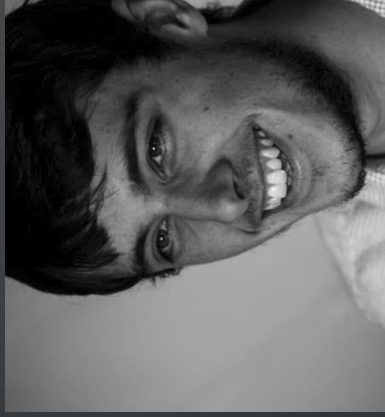
epages



Benjamin Nothdurft

Software Engineer, ePages GmbH

- Wirtschaftsinformatik - Java EE (HS Ulm & Neu-Ulm)
- QA, Testing & Automation (since 2012)
- Speaker & Founder, Softwerkskammer Jena (2016)



Bastian Klein

Software Engineer, ePages GmbH

- Informatik (DHBW Stuttgart & Hewlett-Packard)
- Release Automation, Testing
- Python, Elastic Stack, Java, Selenium

Twitter: [dataduke](#) / [dastianoro](#) / [epagesdevs](#)

Introduction



What are we going to learn?

Why should you listen to us?

epages

Specific Takeaways

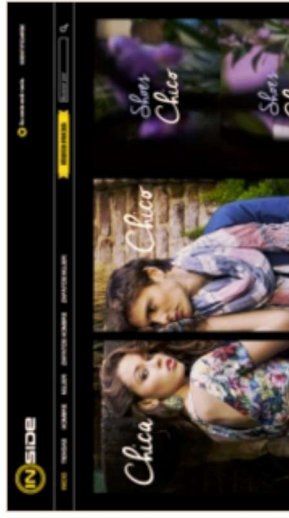
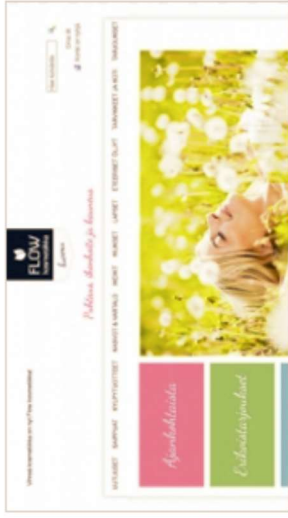
- When do you need to automate?
- How to figure out who your customers are?
- Find your requirements?
- How to find the fitting tools?
- What tools are trending these days?
- How to kickstart things in your project!

Background story

epages



Business Model

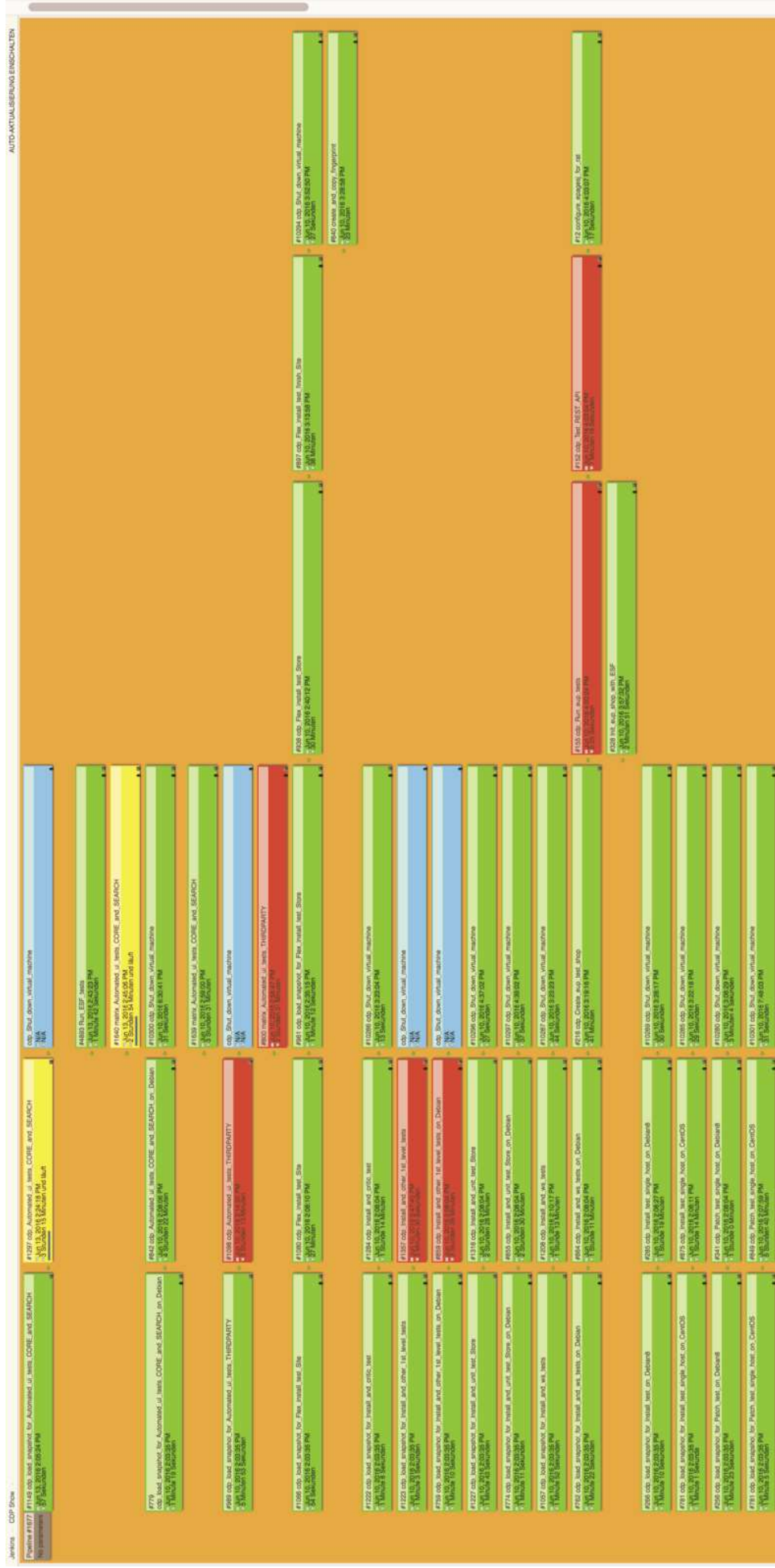


Pipeline Visualization

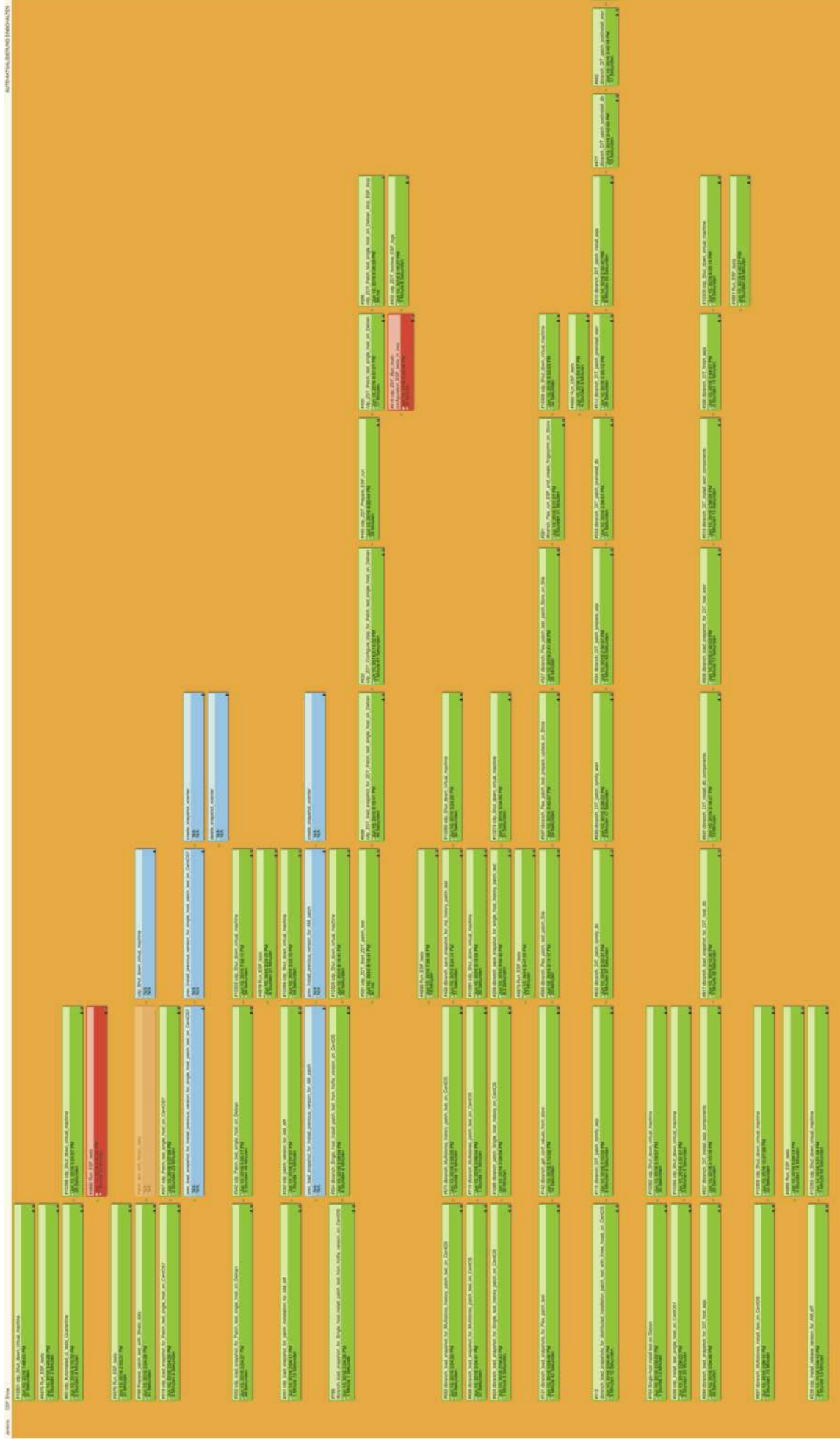
The screenshot displays the Jenkins Pipeline Visualization interface. At the top, the text "AUTO-AKTUALISIERUNG EINSCHALTEN" is visible. The main area shows a sequence of build steps, each represented by a green box with a play button icon on the left and a stop button on the right. The steps are connected by green arrows indicating the flow of the pipeline. The steps are as follows:

- Pipeline #1677**: No parameters, 14 Sekunden.
- #1677 cdp_Tag_MAIN**: Jun 10, 2016 12:05:18 PM, 14 Sekunden.
- #229 cdp_Collect_commit_ids**: Jun 10, 2016 12:05:33 PM, 1,9 Sekunden.
- #230 cdp_load_snapshot_for_Build_patch_files_from_GH**: Jun 10, 2016 12:05:35 PM, 1 Minute 45 Sekunden.
- #246 cdp_Build_patch_files_from_GH**: Jun 10, 2016 12:10:04 PM, 10 Minuten.
- #1411 cdp_Build_packages_from_lagged_MAIN**: Jun 10, 2016 12:21:12 PM, 1 Stunde 42 Minuten.
- #1149 cdp_load_snapshot_for_Automated_ui_tests_CORE_and_SEARCH**: Jun 13, 2016 2:05:24 PM, 37 Sekunden.
- #779 cdp_load_snapshot_for_Automated_ui_tests_CORE_and_SEARCH_on_Debian**: Jun 10, 2016 2:03:35 PM, 1 Minute 19 Sekunden.
- #989 cdp_load_snapshot_for_Automated_ui_tests_THIRDPARTY**: Jun 10, 2016 2:03:35 PM, 5 Minuten 53 Sekunden.
- #1066 cdp_load_snapshot_for_Flex_install_test_Site**: Jun 10, 2016 2:03:35 PM, 54 Sekunden.
- #1222 cdp_load_snapshot_for_install_and_critics_test**: Jun 10, 2016 2:03:35 PM, 1 Minute 6 Sekunden.

Pipeline Visualization



Pipeline Visualization



Test Pyramid

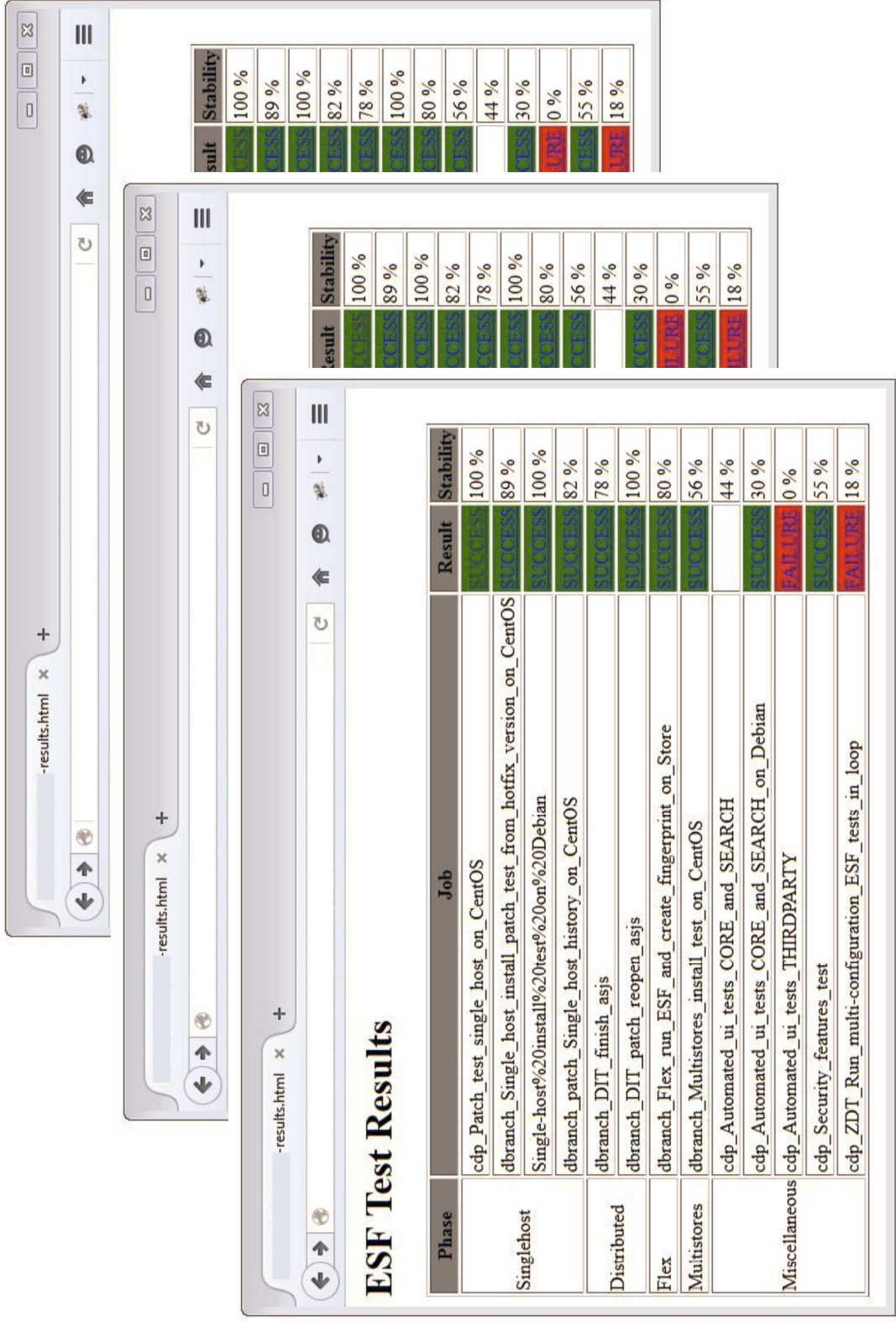


Demo Time

Analyze the test results



Static HTML Pages



Jenkins Job Hierarchy

- Zurück zum Projekt
- Status
- Änderungen
- Console Output
- Build-Informationen anzeigen
- Parameter
- Downstream build view
- Environment Variables
- Embeddable Build Status
- Build Graph
- Vorheriger Build
- Nachfolgender Build

Vor 4 Tage 12 Stunden gestartet
Dauer: 58 Minuten auf [dbranch-vm-install-di-asis-centos-01](#)

Build #593 (09.06.2016 05:04:58)

Subproject Builds

Run_ESF_tests #4846



Keine Änderungen.



Gestartet durch vorgelagertes Projekt [dbranch_DIT_install_wstr_components](#) , Build 615
ursprünglich ausgelöst durch:

- Gestartet durch vorgelagertes Projekt [dbranch_load_snapshot_for_DIT_host_wstr](#) , Build 605
ursprünglich ausgelöst durch:
 - Gestartet durch vorgelagertes Projekt [dbranch_DIT_install_db_components](#) , Build 638
ursprünglich ausgelöst durch:
 - Gestartet durch vorgelagertes Projekt [dbranch_load_snapshot_for_DIT_host_db](#) , Build 614
ursprünglich ausgelöst durch:
 - Gestartet durch vorgelagertes Projekt [dbranch_DIT_install_asis_components](#) , Build 834
ursprünglich ausgelöst durch:
 - Gestartet durch vorgelagertes Projekt [dbranch_load_snapshot_for_DIT_host_asis](#) , Build 681
ursprünglich ausgelöst durch:
 - Gestartet durch vorgelagertes Projekt [dbranch_distributed_installation_test_with_three_hosts_on_CentOS](#) , Build 681
ursprünglich ausgelöst durch:
 - Gestartet durch vorgelagertes Projekt [dbranch_installation_test_wrapper_HEAD](#) , Build 519
ursprünglich ausgelöst durch:
 - Gestartet durch vorgelagertes Projekt [cdp_Build_packages_from_tagged_MAIN](#) , Build 1408
ursprünglich ausgelöst durch:
 - Gestartet durch vorgelagertes Projekt [cdp_Build_patch_files_from_Git](#) , Build 243
ursprünglich ausgelöst durch:
 - Gestartet durch vorgelagertes Projekt [cdp_load_snapshot_for_Build_patch_files_from_Git](#) , Build 227
ursprünglich ausgelöst durch:
 - Gestartet durch vorgelagertes Projekt [cdp_Collect_commit_ids](#) , Build 226
ursprünglich ausgelöst durch:
 - (deeply nested causes)



This run spent:

- 3 Minuten 44 Sekunden waiting in the queue;
- 58 Minuten building on an executor;
- 1 Stunde 2 Minuten total from scheduled to completion.

Jenkins Job Queue & Artifacts

Artifacts of Run_ESF_tests #4880

| Build-Verlauf | Trend |
|-----------------------|------------------|
| #4897 | 13.06.2016 17:01 |
| #4896 | 13.06.2016 15:51 |
| #4895 | 13.06.2016 15:38 |
| #4894 | 13.06.2016 14:53 |
| #4893 | 13.06.2016 14:43 |
| #4892 | 13.06.2016 14:16 |
| #4891 | 13.06.2016 14:15 |
| #4890 | 13.06.2016 14:11 |
| #4889 | 13.06.2016 13:44 |
| #4888 | 13.06.2016 12:40 |
| #4887 | 13.06.2016 12:26 |
| #4886 | 10.06.2016 19:56 |
| #4885 | 10.06.2016 18:09 |
| #4884 | 10.06.2016 17:50 |
| #4883 | 10.06.2016 17:27 |
| #4882 | 10.06.2016 17:24 |
| #4881 | 10.06.2016 16:40 |

 [esf / esf-epages6-1.15.0-SNAPSHOT / log / 20160610T140043701Z /](#) 

-  [esf-test-reports/com/epages/cartridges/de_epages](#)
-  [testNG-output](#)
-  [browser.log](#) 4.54 MB  [anzeigen](#)
-  [esf-debug.log](#) 29.57 KB  [anzeigen](#)
-  [esf-errors.log](#) 0 B  [anzeigen](#)
-  [index.html](#) 21.31 KB  [anzeigen](#)
-  [style.css](#) 163 B  [anzeigen](#)

 [\(Alle Dateien als ZIP-Archiv herunterladen\)](#)

Test Report Index

| Time Info | | Quantity | | Configuration | |
|-----------|--------------------------|----------|----|---------------|---|
| Start | 2016-06-10T14:00:43.701Z | All | 38 | URL | |
| End | 2016-06-10T15:23:38.125Z | Failed | 20 | Shop | quarantine-test-1, quarantine-test-2, quarantine-test-3, quarantine-test-4, quarantine-test-5, quarantine-test-6, quarantine-test-7, quarantine-test-8, quarantine-test-9, quarantine-test-10, quarantine-test-11, quarantine-test-12, quarantine-test-13, quarantine-test-14, quarantine-test-15, quarantine-test-16, quarantine-test-17, quarantine-test-18, quarantine-test-19, quarantine-test-20, quarantine-test-21, quarantine-test-22 |
| Duration | 01:22:54 | Skipped | 9 | | |
| | | Passed | 9 | Test groups | CORE,SEARCH,THIRDPARTY |

Results

| Index | Test Class | Test Method | Result | Test Report | Runtime |
|-------|--|--|---------|------------------|---------|
| 22 | com.epages.cartridges.de_epages.paybox.tests.mbo.PayboxTransactions | payboxTransactionTests | FAILURE | test-report.html | 188 |
| 14 | com.epages.cartridges.de_epages.ebay.tests.EbayTest | ebayConfigurationBBOTest | FAILURE | test-report.html | 85 |
| 25 | com.epages.cartridges.de_epages.payment.free.tests.MobileFreeCheckoutTest | mobileCheckoutWithFree | FAILURE | test-report.html | 78 |
| 26 | com.epages.cartridges.de_epages.saferpay.tests.SaferpayBusinessCheckoutTest | checkoutWithSaferpayBusiness | FAILURE | test-report.html | 189 |
| 1 | com.epages.cartridges.de_epages.saferpay.tests.SaferpayDebitCheckoutTest | checkoutWithSaferpayDebit | FAILURE | test-report.html | 191 |
| 2 | com.epages.cartridges.de_epages.saferpay.tests.SaferpayEcommerceCheckoutTest | checkoutWithSaferpayEcommerceCreditCard | FAILURE | test-report.html | 184 |
| 3 | com.epages.cartridges.de_epages.saferpay.tests.SaferpayEcommerceCheckoutTest | checkoutWithSaferpayEcommerceDirectDebit | FAILURE | test-report.html | 167 |
| 31 | com.epages.cartridges.de_epages.docdata.tests.DocdataTest | testVisaBeforePayment | FAILURE | test-report.html | 107 |
| 29 | com.epages.cartridges.de_epages.docdata.tests.DocdataTest | testIdealBeforePayment | FAILURE | test-report.html | 105 |
| 30 | com.epages.cartridges.de_epages.docdata.tests.DocdataTest | testMastercardBeforePayment | FAILURE | test-report.html | 108 |
| 33 | com.epages.cartridges.de_epages.docdata.tests.DocdataTest | testClassicBankTransferBeforePayment | FAILURE | test-report.html | 83 |
| 35 | com.epages.cartridges.de_epages.docdata.tests.DocdataTest | testCancelAfterPayment | FAILURE | test-report.html | 145 |
| 34 | com.epages.cartridges.de_epages.docdata.tests.DocdataTest | testClassicIdealBeforePayment | FAILURE | test-report.html | 68 |
| 32 | com.epages.cartridges.de_epages.docdata.tests.DocdataTest | testCancelBeforePayment | FAILURE | test-report.html | 108 |
| 28 | com.epages.cartridges.de_epages.docdata.tests.DocdataTest | testBankTransferBeforePayment | FAILURE | test-report.html | 112 |
| 5 | com.epages.cartridges.de_epages.paypalplus.tests.PaypalplusPaymentTest | testMBOPaymentSetup | FAILURE | test-report.html | 92 |
| 21 | com.epages.cartridges.de_epages.billsafe.tests.BillSafeCheckoutTest | checkoutWithBillSafe | FAILURE | test-report.html | 288 |
| 23 | com.epages.cartridges.de_epages.order.tests.NewCheckoutAsCreatedCustomerTest | createUserAndCheckout | FAILURE | test-report.html | 147 |
| 19 | com.epages.cartridges.de_epages.shop.tests.DashboardWidgetsTest | dashboardWidgetsTest | FAILURE | test-report.html | 96 |
| 36 | com.epages.cartridges.de_epages.paypal.tests.PayPalIntegralEvolutionTest | checkoutWithPayPalIntegralEvolution | FAILURE | test-report.html | 229 |
| 17 | com.epages.cartridges.de_epages.ebay.tests.EbayTest | finishProductDetailsWithWizard | SKIP | test-report.html | 0 |
| 18 | com.epages.cartridges.de_epages.ebay.tests.EbayTest | csvImportExport | SKIP | test-report.html | 0 |
| 15 | com.epages.cartridges.de_epages.ebay.tests.EbayTest | ebayConfigurationMBOTest | SKIP | test-report.html | 0 |
| 16 | com.epages.cartridges.de_epages.ebay.tests.EbayTest | addProductToEbayFixedItemNumber | SKIP | test-report.html | 0 |
| 7 | com.epages.cartridges.de_epages.paypalplus.tests.PaypalplusPaymentTest | testMultistepOrderBeforePayment | SKIP | test-report.html | 0 |
| 10 | com.epages.cartridges.de_epages.paypalplus.tests.PaypalplusPaymentTest | testClassicOrderAfter | SKIP | test-report.html | 0 |
| 8 | com.epages.cartridges.de_epages.paypalplus.tests.PaypalplusPaymentTest | testMobileOrderBeforePayment | SKIP | test-report.html | 0 |
| 6 | com.epages.cartridges.de_epages.paypalplus.tests.PaypalplusPaymentTest | testFirstWithoutOrderVerification | SKIP | test-report.html | 0 |
| 9 | com.epages.cartridges.de_epages.paypalplus.tests.PaypalplusPaymentTest | testMultistepCancelPaymentOrderBefore | SKIP | test-report.html | 0 |
| 11 | com.epages.cartridges.de_epages.remotesearch.tests.FilterSearchTest | filterByNew | SUCCESS | test-report.html | 13 |
| 20 | com.epages.cartridges.de_epages.forum.tests.BlogTest | createBlogAndBlogpostTest | SUCCESS | test-report.html | 155 |
| 37 | com.epages.cartridges.de_epages.guestbook.tests.GuestBookTest | createGuestBookTest | SUCCESS | test-report.html | 105 |
| 27 | com.epages.cartridges.de_epages.docdata.tests.DocdataTest | testPaymentSetup | SUCCESS | test-report.html | 160 |
| 38 | com.epages.cartridges.de_epages.presentation.tests.CustomFormTest | createCustomFormTest | SUCCESS | test-report.html | 149 |
| 4 | com.epages.cartridges.de_epages.paypalplus.tests.PaypalplusPaymentTest | testBBOPaymentSetup | SUCCESS | test-report.html | 31 |
| 12 | com.epages.cartridges.de_epages.shop.tests.StorefrontLinkTest | testGoToWebsite | SUCCESS | test-report.html | 31 |
| 13 | com.epages.cartridges.de_epages.shop.tests.StorefrontLinkTest | testOpenMobileView | SUCCESS | test-report.html | 7 |
| 24 | com.epages.cartridges.de_epages.shop.tests.SEOCockpitTest | seoCockpitTest | SUCCESS | test-report.html | 192 |

Individual HTML Test Report

Test Report

| Index | Test Class | Test Method | Result |
|-------|---|-------------------------|---------|
| 22 | com.epages.cartridges.de_epages.paybox.tests.mbo.PayboxTransactions | paybox.TransactionTests | FAILURE |

[Info](#) [Stacktrace](#) [Logfile](#) [Screenshots](#)

Test Result

| | |
|---------|--|
| Status | FAILURE |
| Message | Timed out after 30 seconds waiting for presence of element located by: By.id: NUMERO_CARTE Build info: version: '2.52.0', revision: '4c2593cfc3689a7fcd7be52549167e5ccc93ad28', time: '2016-02-11 11:22:43' System info: host: 'ci-vm-ui-test-003', ip: '127.0.1.1', os.name: 'linux', os.arch: 'amd64', os.version: '3.13.0-43-generic', java.version: '1.8.0_45-internal' Driver info: org.openqa.selenium.support.events.EventFiringWebDriver |
| Runtime | 188 |

Configuration

| | |
|---------|---|
| Browser | FIREFOX |
| URL | |
| Shop | quarantine-test-21,quarantine-test-20,quarantine-test-19,quarantine-test-18,quarantine-test-17,quarantine-test-16,quarantine-test-15,quarantine-test-14,quarantine-test-13,quarantine-test-12,quarantine-test-11,quarantine-test-10,quarantine-test-9,quarantine-test-8,quarantine-test-7,quarantine-test-6,quarantine-test-5,quarantine-test-4,quarantine-test-3,quarantine-test-2,quarantine-test-1 |

Delivery Pipeline in Jenkins

| Feature Team | Release Automation Team |
|--|---|
| <ul style="list-style-type: none">- No idea where to find documents- No experience with Jenkins- Very limited time for integration | <ul style="list-style-type: none">- Inspect a lot of Jenkins jobs- All documents at different places- Results stored only for 30 days |

Daily Fight

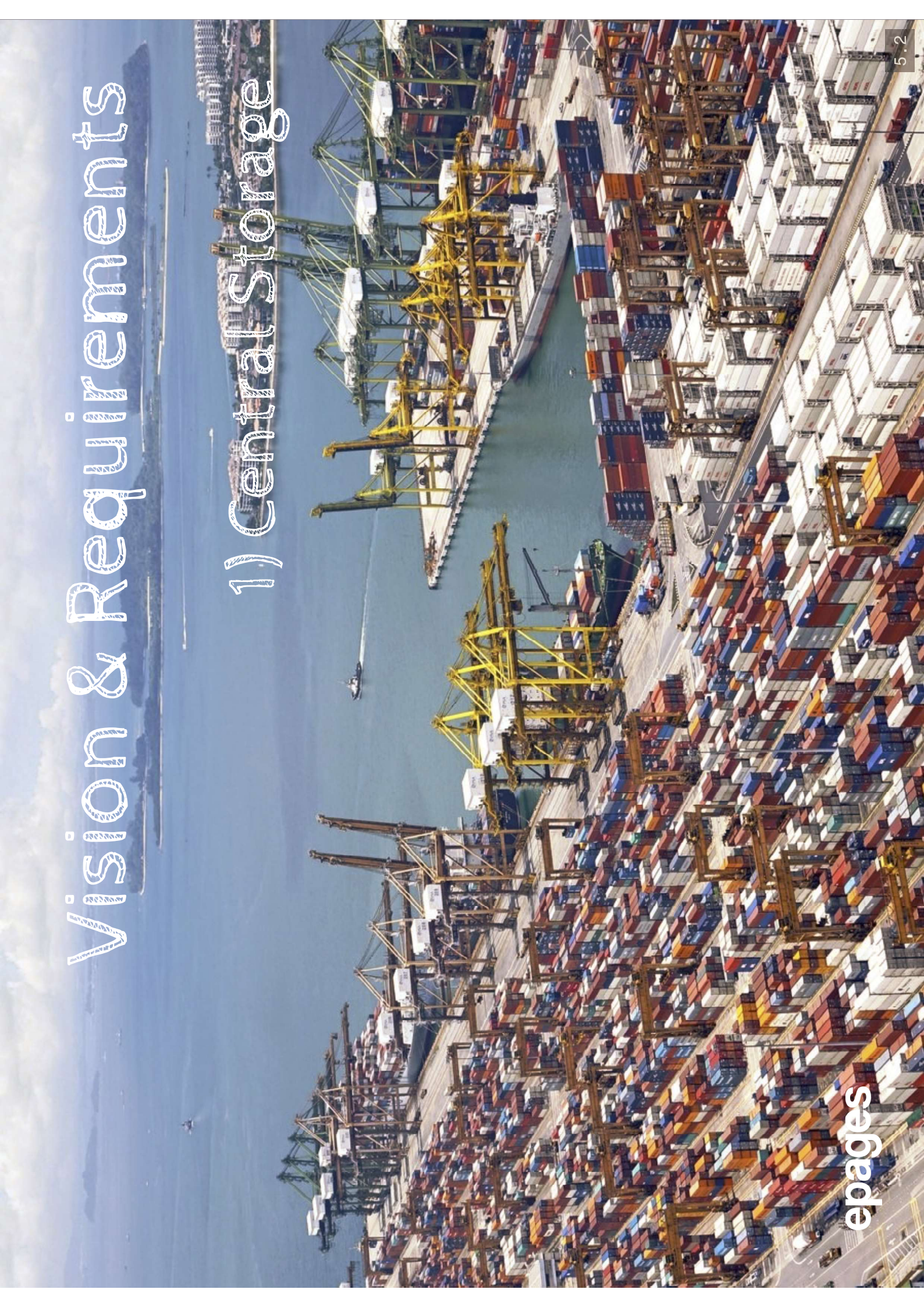




Solution Approach

Vision & Requirements

1) Central storage



Vision & Requirements

1) Central Storage

2) Simple & Maintainable



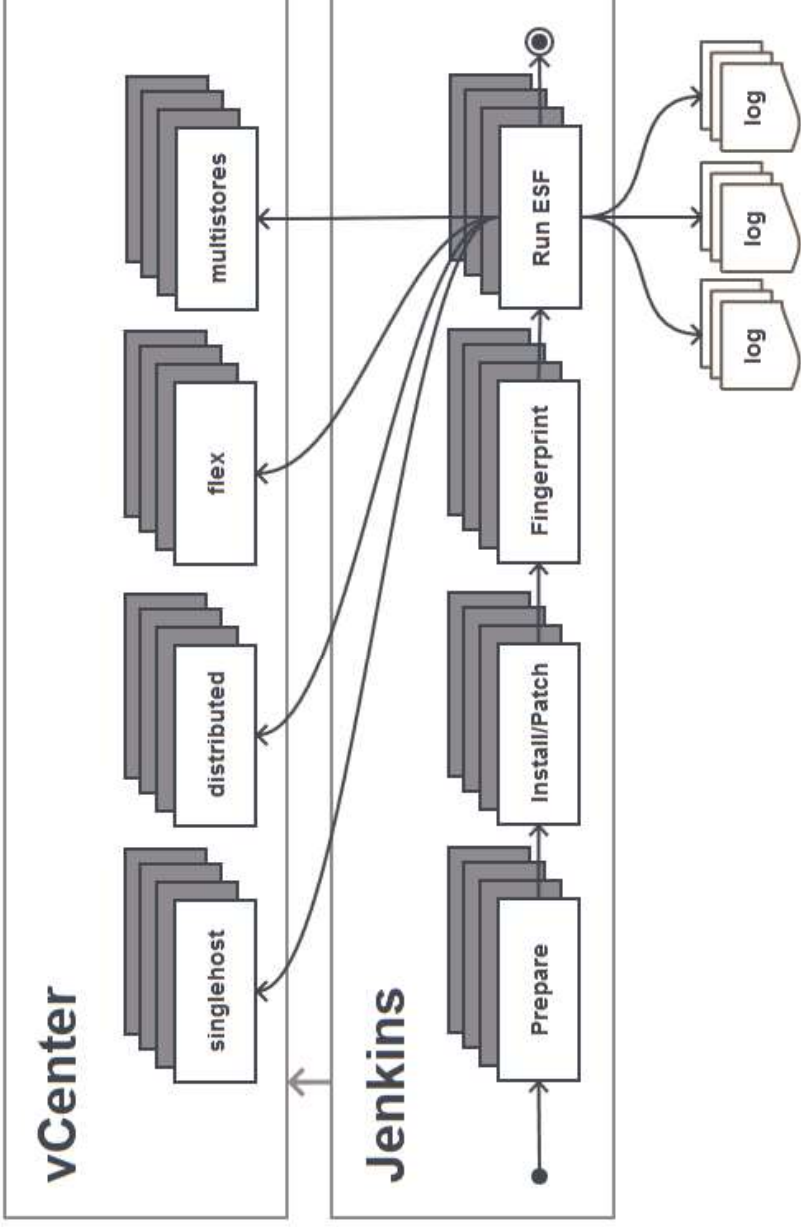
Vision & Requirements

1) Central Storage

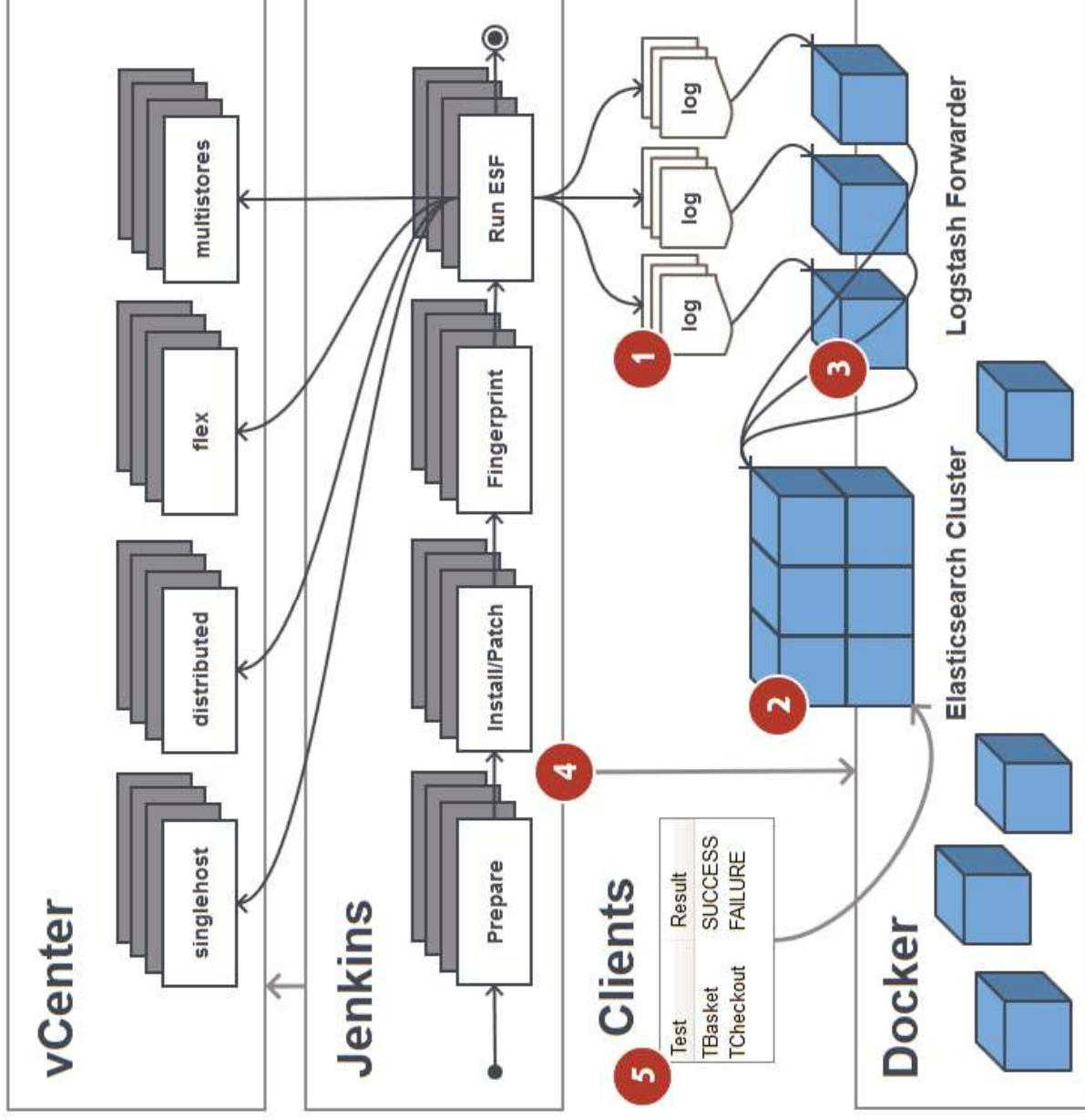
2) Simple & Maintainable

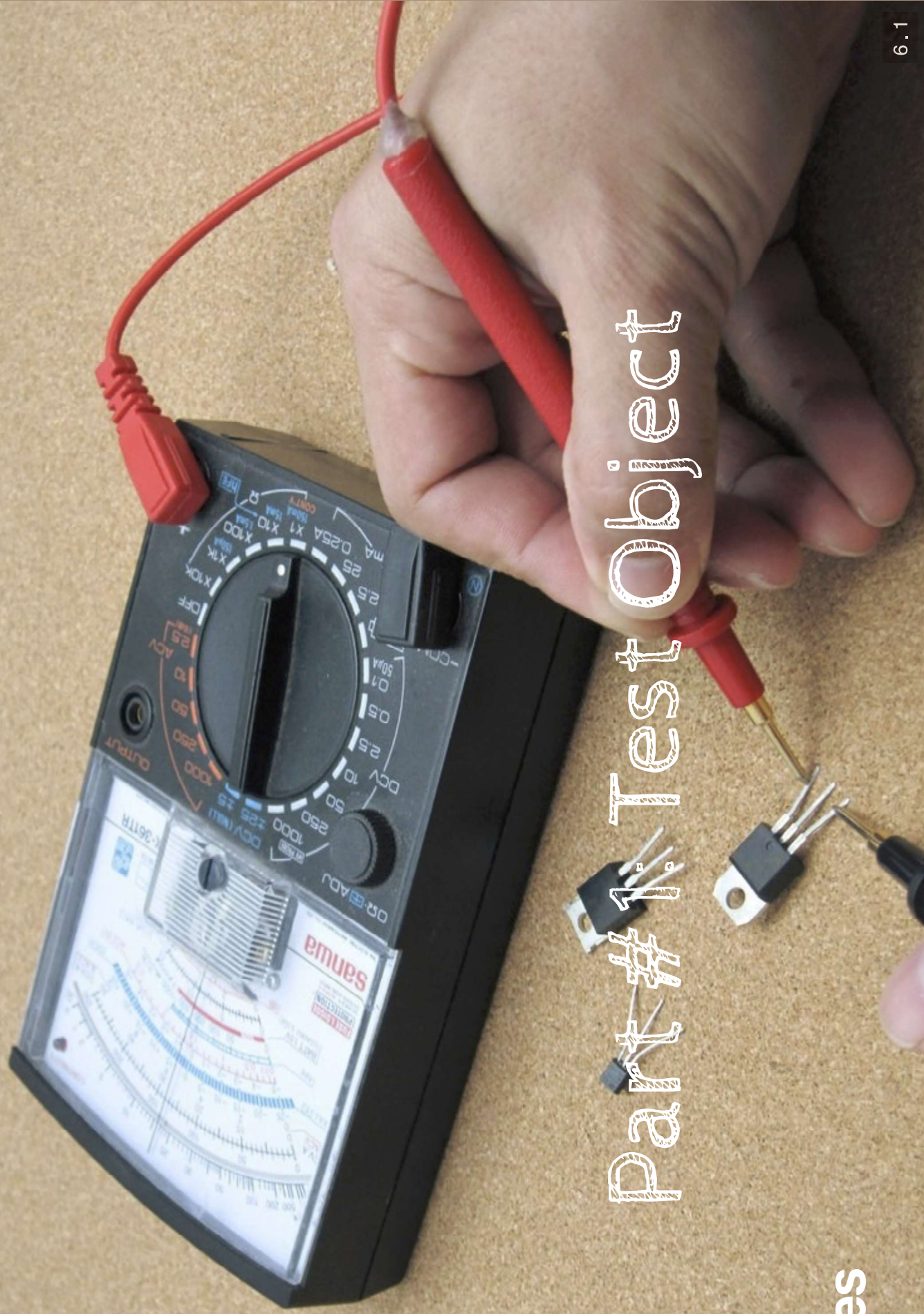
3) Easy to view via website

Current State



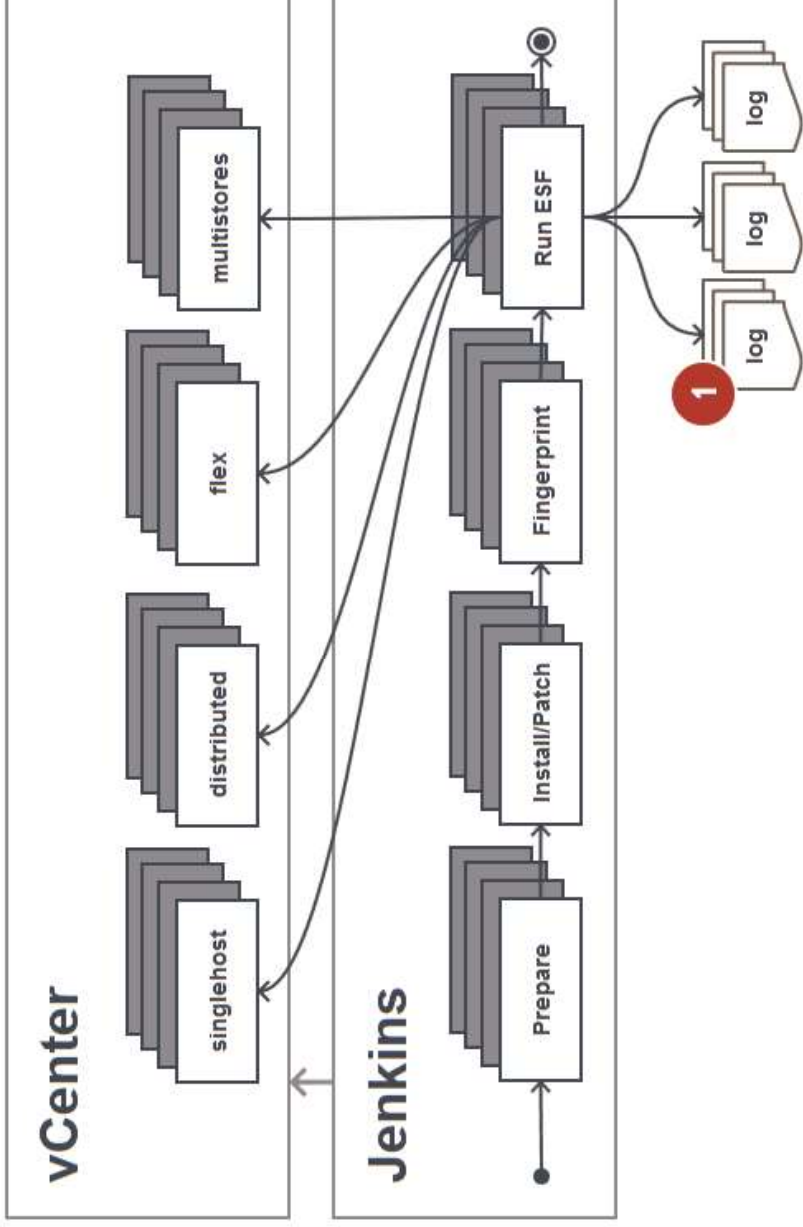
Solution Blueprint





Part #1 Test Object

Test Aggregation Workflow



Test Object from Test Suite

```
{
  "browser": "firefox",
  "timestamp": "2016-06-13T19:23:32.227Z",
  "pos": "1",
  "result": "FAILURE",
  "test": "EbayTest.ebayConfigurationBBOTest",
  "class": "com.epages.cartridges.de_epages.ebay.tests.EbayTest",
  "method": "ebayConfigurationBBOTest",
  "runtime": "67",
  "team": "ePages6",
  "test_url": "/20160613T192332227Z/esf-test-reports/
com/epages/cartridges/de_epages/ebay/tests/
EbayTest/ebayConfigurationBBOTest/test-report.html",
  "stacktrace": "java.lang.NullPointerException
at com.epages.cartridges.de_epages.ebay.tests.EbayTest.ebayConfigurationBBOTest(EbayTest.
at sun.reflect.NativeMethodAccessorImpl.invoke0(Native Method)
at sun.reflect.NativeMethodAccessorImpl.invoke(NativeMethodAccessorImpl.java:62)
at sun.reflect.DelegatingMethodAccessorImpl.invoke(DelegatingMethodAccessorImpl.java:43)
at java.lang.reflect.Method.invoke(Method.java:498)
at org.testng.internal.MethodInvocationHelper.invokeMethod(MethodInvocationHelper.java:86
at org.testng.internal.Invoker.invokeMethod(Invoker.java:643)
at org.testng.internal.Invoker.invokeTestMethod(Invoker.java:820)
at org.testng.internal.Invoker.invokeTestMethods(Invoker.java:1128)"
}
```

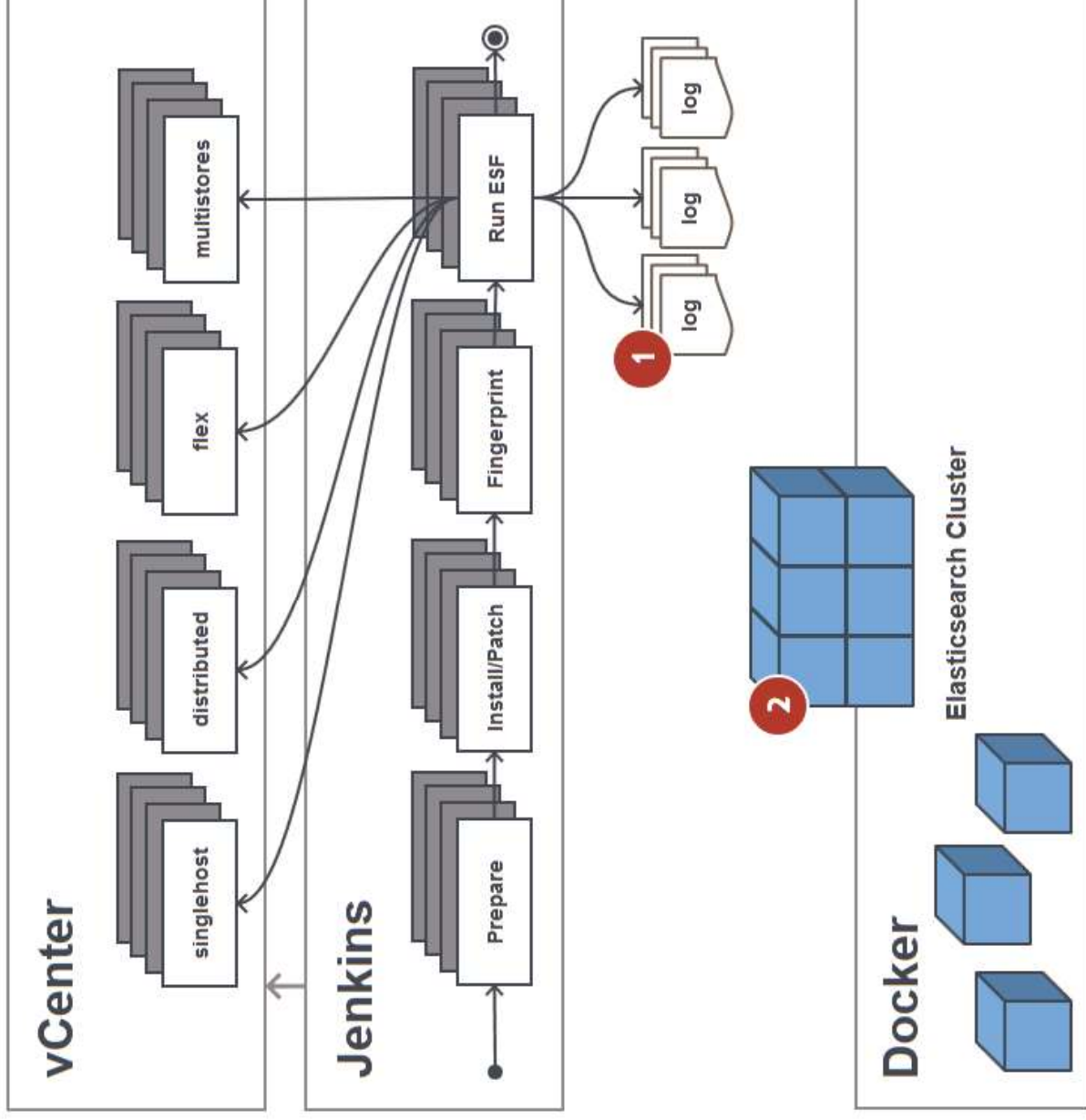
Test Object in Elasticsearch

```
{
  "epages_version": "6.17.48",
  "epages_repo_id": "6.17.48/2016.05.19-00.17.26",
  "env_os": "centos",
  "env_identifier": "distributed_three_hosts",
  "env_type": "install",
  "browser": "firefox",
  "timestamp": "20160519T011223091Z",
  "pos": "3",
  "result": "FAILURE",
  "test": "DigitalTaxmatrixBasketTest.testDigitalTaxmatrixBasket",
  "class": "com.epages.cartridges.de_epages.tax.tests.DigitalTaxmatrixBasketTest",
  "method": "testDigitalTaxmatrixBasket",
  "runtime": "275",
  "report_url": "http://myserver.epages.de:8080/job/Run_ESF_tests/3778/artifact/esf/
  esf-epages6-1.15.0-SNAPSHOT/log/20160519T001726091Z/
  esf-test-reports/com/epages/cartridges/de_epages/tax/tests/DigitalTaxmatrixBasketTest/
  testDigitalTaxmatrixBasket/test-report.html",
  "stacktrace": "org.openqa.selenium.TimeoutException:
Timed out after 30 seconds waiting for presence of element located by: By.className:
Saved Build info: version: '2.47.1', System info: host: 'ci-vm-ui-test-004',
ip: '127.0.1.1', os.name: 'Linux', os.arch: 'amd64', os.version: '3.13.0-43-generic', java
org.openqa.selenium.support.events.EventFiringWebDriver at
org.openqa.selenium.support.ui.WebDriverWait.timeoutException(WebDriverWait.java:80) at
org.openqa.selenium.support.ui.FluentWait.until(FluentWait.java:229) at
com.epages.esf.controller.ActionBot.waitFor(ActionBot.java:491) at com.epages.esf.contrc
com.epages.cartridges.de_epages.coupon.pageobjects.mbo.ViewCouponCodes.createManualCoupc
com.epages.cartridges.de_epages.tax.tests.DigitalTaxmatrixBasketTest.setupCoupon(Digital
com.epages.cartridges.de_epages.tax.tests.DigitalTaxmatrixBasketTest.testDigitalTaxmatri
}
```




Part #2: Elasticsearch

Test Aggregation Workflow



to-elasticsearch/Dockerfile

```
# Use official Elasticsearch image.
FROM elasticsearch:2.2.3

#####
# Install Jinja2 #
#####

ENV JINJA_SCRIPT="/render_jinja_template.py" \
    REPO_API_PATH="https://api.github.com/repos/gh-acc/gh-repo/contents/scripts/templating" \
    REPO_PROD_BRANCH="master"

# Install packages and clean-up
RUN apt-get update && apt-get install -y curl python-setuptools && \
    easy_install Jinja2 && \
    apt-get -y clean && \
    rm -rf /var/lib/apt/lists/*

# Add jinja templating script from repo
RUN curl --retry 5 -H "Authorization: token ${REPO_ACCESS_TOKEN}" \
    -H 'Accept: application/vnd.github.v3.raw' \
    -o ${JINJA_SCRIPT} -L ${REPO_API_PATH}${JINJA_SCRIPT}?ref=${REPO_PROD_BRANCH} && \
    chown elasticsearch:elasticsearch ${JINJA_SCRIPT} && \
    chmod +x ${JINJA_SCRIPT}
```

...

to-elasticsearch/Dockerfile

```
#####  
# Elasticsearch #  
#####  
  
ENV ES_PATH="/usr/share/elasticsearch" \  
    ES_HTTP_BASIC="https://github.com/Asquera/elasticsearch-http-basic/releases/download/v1.5.  
  
RUN $ES_PATH/bin/plugin -install mobz/elasticsearch-head  
  
RUN mkdir -p $ES_PATH/plugins/http-basic && \  
    cd $ES_PATH/plugins/http-basic && \  
    wget $ES_HTTP_BASIC  
  
ENV ES_CONFIG_VOL="/usr/share/elasticsearch/config" \  
    ES_DATA_VOL="/usr/share/elasticsearch/data" \  
    ES_LOGS_VOL="/usr/share/elasticsearch/logs"  
  
COPY config/ ${ES_CONFIG_VOL}/  
RUN chown -R elasticsearch:elasticsearch ${ES_CONFIG_VOL}  
  
VOLUME [ "${ES_CONFIG_VOL}", "${ES_LOGS_VOL}" ]  
  
RUN rm /docker-entrypoint.sh  
  
COPY docker-entrypoint.sh /  
RUN chown elasticsearch:elasticsearch /docker-entrypoint.sh && \  
    chmod +x /docker-entrypoint.sh  
ENTRYPOINT [ "/docker-entrypoint.sh" ]  
  
CMD [ "elasticsearch" ]
```

to-elasticsearch/docker-entrypoint.sh

```
#!/bin/bash

set -e

# Add elasticsearch as command if needed
if [ "${1:0:1}" = '-' ]; then
    set -- elasticsearch "$@"
fi

# Drop root privileges if we are running elasticsearch
if [ "$1" = 'elasticsearch' ]; then

    # Change the ownership of /usr/share/elasticsearch/data to elasticsearch
    chown -R elasticsearch:elasticsearch ${ES_CONFIG_VOL} ${ES_DATA_VOL} ${ES_LOGS_VOL}

    # Find env file in docker
    ES_ENV_PATH=$( find "${ES_CONFIG_VOL}" -maxdepth 3 -iname "${ES_ENV}" )

    # Render jinja templates of elasticsearch.yaml and logging.yaml
    python ${JINJA_SCRIPT} -f "${ES_ENV_PATH}" \
        -t "${ES_CONFIG_VOL}"/elasticsearch.yml.j2 \
          "${ES_CONFIG_VOL}"/logging.yml.j2

    set -- gosu elasticsearch "${@}"
fi

# As argument is not related to elasticsearch,
# then assume that user wants to run his own process,
# for example a `bash` shell to explore this image
exec "${@}"
```

./config/elasticsearch.yml.12

```
#####  
# Cluster #  
#####  
  
# Set the cluster name  
cluster.name: {{ CLUSTER_NAME }}  
  
#####  
# Node #  
#####  
  
# Prevent Elasticsearch from choosing a new name on every startup.  
node.name: {{ NODE_NAME }}  
  
# Allow this node to be eligible as a master node  
node.master: {{ NODE_MASTER }}  
  
# Allow this node to store data  
node.data: {{ NODE_DATA }}  
  
#####  
# Path #  
#####  
  
path.config: /usr/share/elasticsearch/config  
path.plugins: /usr/share/elasticsearch/plugins  
path.data: /usr/share/elasticsearch/data  
path.logs: /usr/share/elasticsearch/logs  
path.work: /usr/share/elasticsearch/work
```

./config/elasticsearch.yml.j2

```
#####  
# Network #  
#####  
  
network.bind_host: 0.0.0.0  
network.publish_host: 0.0.0.0  
transport.tcp.port: 9300  
http.port: 9200  
http.enabled: true  
  
#####  
# HTTP Module #  
#####  
  
http.cors.enabled: {{ HTTP_ENABLED }}  
http.cors.allow-origin: {{ HTTP_ALLOW_ORIGIN }}  
http.cors.allow-methods: {{ HTTP_ALLOW_METHODS }}  
http.cors.allow-headers: {{ HTTP_ALLOW_HEADERS }}  
  
#####  
# HTTP Basic Plugin #  
#####  
  
http.basic.enabled: true  
http.basic.user: {{ ES_USER }}  
http.basic.password: {{ ES_PASSWORD }}
```

./config/elasticsearch.yml.j2

```
#####  
# Slowlog Module #  
#####  
  
# Set threshold for shard level query execution logging  
index.search.slowlog.threshold.query.warn : 10s  
index.search.slowlog.threshold.query.info : 5s  
index.search.slowlog.threshold.query.debug : 2s  
index.search.slowlog.threshold.query.trace : 500ms  
  
# Set threshold for shard level fetch phase logging  
index.search.slowlog.threshold.fetch.warn : 1s  
index.search.slowlog.threshold.fetch.info : 800ms  
index.search.slowlog.threshold.fetch.debug : 500ms  
index.search.slowlog.threshold.fetch.trace : 200ms  
  
# Set threshold for shard level index logging  
index.indexing.slowlog.threshold.index.warn : 10s  
index.indexing.slowlog.threshold.index.info : 5s  
index.indexing.slowlog.threshold.index.debug : 2s  
index.indexing.slowlog.threshold.index.trace : 500ms  
  
#####  
# GC Logs #  
#####  
  
# Set threshold for young garbage collection logging  
monitor.jvm.gc.young.warn : 1000ms  
monitor.jvm.gc.young.info : 700ms  
monitor.jvm.gc.young.debug : 400ms
```


./config/env-to-master-01.list

The variables used for rendering of jinja templates.

```
#####  
# env variables #  
#####
```

```
ES_ENV  
ES_HEAP_SIZE
```

```
#####  
# elasticsearch.yml #  
#####
```

```
CLUSTER_NAME=to-elasticsearch  
NODE_NAME=to-es-master-01  
NODE_MASTER=true  
NODE_DATA=true  
HTTP_ENABLED=true  
HTTP_ALLOW_ORIGIN=/*/  
HTTP_ALLOW_METHODS=OPTIONS, HEAD, GET, POST, PUT, DELETE  
HTTP_ALLOW_HEADERS=Authorization  
ES_USER  
ES_PASSWORD
```

```
#####  
# logging.yml #  
#####
```

```
LOG_LEVEL=INFO
```

./scripts/render_jinja_template.py

```
usage: render.py [-h] [-v] [-e ENV [ENV ...]] [-f FILES [FILES ...]]
               -t TEMPLATES [TEMPLATES ...] [-d DEST]
```

script to render jinja template with env variables and output rendered file.

invocation:

```
render_jinja_template.py -v
-t <filename>.<ext>.j2
-e <key> <key>=<value>
-f <env.list>
-d </dest/dir>

render_jinja_template.py --verbose
--template <filename>.<ext>.j2
--env <key> <key>=<value>
--env-file <env.list>
--dest </dest/dir>
```

to-elasticsearch/circle.yml

```
machine:
  services:
    - docker
  environment:
    # Test uses a dedicated docker container.
    TEST_TO_MASTER: "to-es-master"
    TEST_TO_MASTER_ENV: "env-to-es-master-01.list"
    # Docker run options are set to detach to background and share network addresses from host
    LS_DOCKER_REMOVE: false
    LS_DOCKER_DETACH: true
    # Docker build image.
    ES_IMAGE_NAME: "epages/to-elasticsearch"
    ES_IMAGE_TAG: ${CIRCLE_BRANCH//\//-}
    # Host connection details.
    ES_HOST_URL: "http://0.0.0.0"
    ES_HOST_HTTP: 9200
    ES_HOST_TCP: 9300
    # Test connection times.
    SLEEP_BEFORE_TESTING: 15
    # Git merge script is needed for auto-merging dev to master branch.
    MERGE_SCRIPT: "merge-to.sh"
    MERGE_SCRIPT_URL_PREFIX: "https://raw.githubusercontent.com/epages-de/repo/master/scripts/"
    GIT_UPSTREAM_URL: "git@github.com:epages-de/to-elasticsearch.git"
    GIT_UPSTREAM_BRANCH_MASTER: "master"
    GIT_UPSTREAM_BRANCH_PRODUCTION: "stable"
```

...

to-elasticsearch/circle.yml

```
..
dependencies:
  cache_directories:
    - "~/docker"
  override:
    # Docker environment used.
    - docker info
    # Load cached images, if available.
    - if [[ -e ~/docker/image.tar ]]; then docker load --input ~/docker/image.tar; fi
    # Build our image.
    - ./build.sh
    # Save built image into cache.
    - mkdir -p ~/docker; docker save ${ES_IMAGE_NAME}:${ES_IMAGE_TAG} > ~/docker/image.tar

test:
  override:
  ..
```

to-elasticsearch/circle.yml

```
..
test:
  override:
    - |
      printf "\n%s\n\n" "+++ Begin test of docker container [${TEST_TO_MASTER}] +++"
      export ES_DOCKER_CONTAINER="${TEST_TO_MASTER}"
      export ES_ENV="${TEST_TO_MASTER_ENV}"
      export ES_CONFIG="/tmp/${TEST_TO_MASTER}/config"
      export ES_DATA="/tmp/${TEST_TO_MASTER}/data"
      export ES_LOGS="/tmp/${TEST_TO_MASTER}/logs"
      mkdir -v -p ${ES_CONFIG} ${ES_DATA} ${ES_LOGS}
      cp -v -r config/* ${ES_CONFIG}/
      # Fire up our container for testing.
      ./start.sh; exit $?
      # Test the access to our Elasticsearch instance.
      sleep ${SLEEP_BEFORE_TESTING}; curl --retry 5 -u ${TEST_ES_USER}:${TEST_ES_PASSWORD} "
      # Stop running container.
      ./stop.sh; exit $?
      # Test our deployment script as well.
      export ES_DOCKER_CONTAINER="${TEST_TO_MASTER}-production"
      ./deploy.sh
      sleep ${SLEEP_BEFORE_TESTING}; curl --retry 5 -u "${ES_USER}:${ES_PASSWORD}" "${ES_HOES}
      printf "\n%s\n\n" "+++ End test of docker container [${TEST_TO_MASTER}] +++"

  post:
    - |
      printf "\n%s\n\n" "=== Archive artifacts of [${TEST_TO_MASTER}] ==="
      sudo mv -v -f "/tmp/${TEST_TO_MASTER}" "${CIRCLE_ARTIFACTS}/"

deployment:
  ..
```

to-elasticsearch/circle.yml

```
..
deployment:
  dev_actions:
    branch: dev
    commands:
      # Push image to Docker Hub.
      - docker login -u "${DOCKER_LOGIN_USERNAME}" -p "${DOCKER_LOGIN_PASSWORD}" -e "${DOCKER_
      - docker push "${ES_IMAGE_NAME}:${ES_IMAGE_TAG}"
      # Merge tested commit into master.
      - wget -O "/tmp/${MERGE_SCRIPT}" "${MERGE_SCRIPT_URL_PREFIX}/${MERGE_SCRIPT}" && chmod 7
      - /tmp/${MERGE_SCRIPT} -c "${CIRCLE_SHA1}" -e "${CIRCLE_BRANCH}" -t "${GIT_UPSTREAM_BRA
  master_actions:
    branch: master
    commands:
      # Push image to Docker Hub.
      - docker login -u "${DOCKER_LOGIN_USERNAME}" -p "${DOCKER_LOGIN_PASSWORD}" -e "${DOCKER_
      - docker push "${ES_IMAGE_NAME}:${ES_IMAGE_TAG}"
      # Merge tested commit into stable.
      - wget -O "/tmp/${MERGE_SCRIPT}" "${MERGE_SCRIPT_URL_PREFIX}/${MERGE_SCRIPT}" && chmod 7
      - /tmp/${MERGE_SCRIPT} -c "${CIRCLE_SHA1}" -e "${CIRCLE_BRANCH}" -t "${GIT_UPSTREAM_BRA
  stable_actions:
    branch: stable
    commands:
      # Push image to Docker Hub.
      - docker login -u "${DOCKER_LOGIN_USERNAME}" -p "${DOCKER_LOGIN_PASSWORD}" -e "${DOCKER_
      - docker push "${ES_IMAGE_NAME}:${ES_IMAGE_TAG}"
      # Tag with 'latest' and push image to Docker Hub.
      - docker tag "${ES_IMAGE_NAME}:${ES_IMAGE_TAG}" "${ES_IMAGE_NAME}:latest"
      - docker push "${ES_IMAGE_NAME}:latest"
```

to-elasticsearch/deploy.sh

```
#!/bin/bash

export SCRIPT_DIR=$(dirname "$0")

# Half of the available RAM should be used for ES directly. The other half can
# be consumed by Lucene (via the OS' filesystem cache).
export ES_HEAP_SIZE=4g

# Do we have an Image and Tag to be used?
if [[ -z "${ES_IMAGE_NAME}" ]] ; then
    echo 'Variable ES_IMAGE_NAME is not set.'
    exit 1
fi

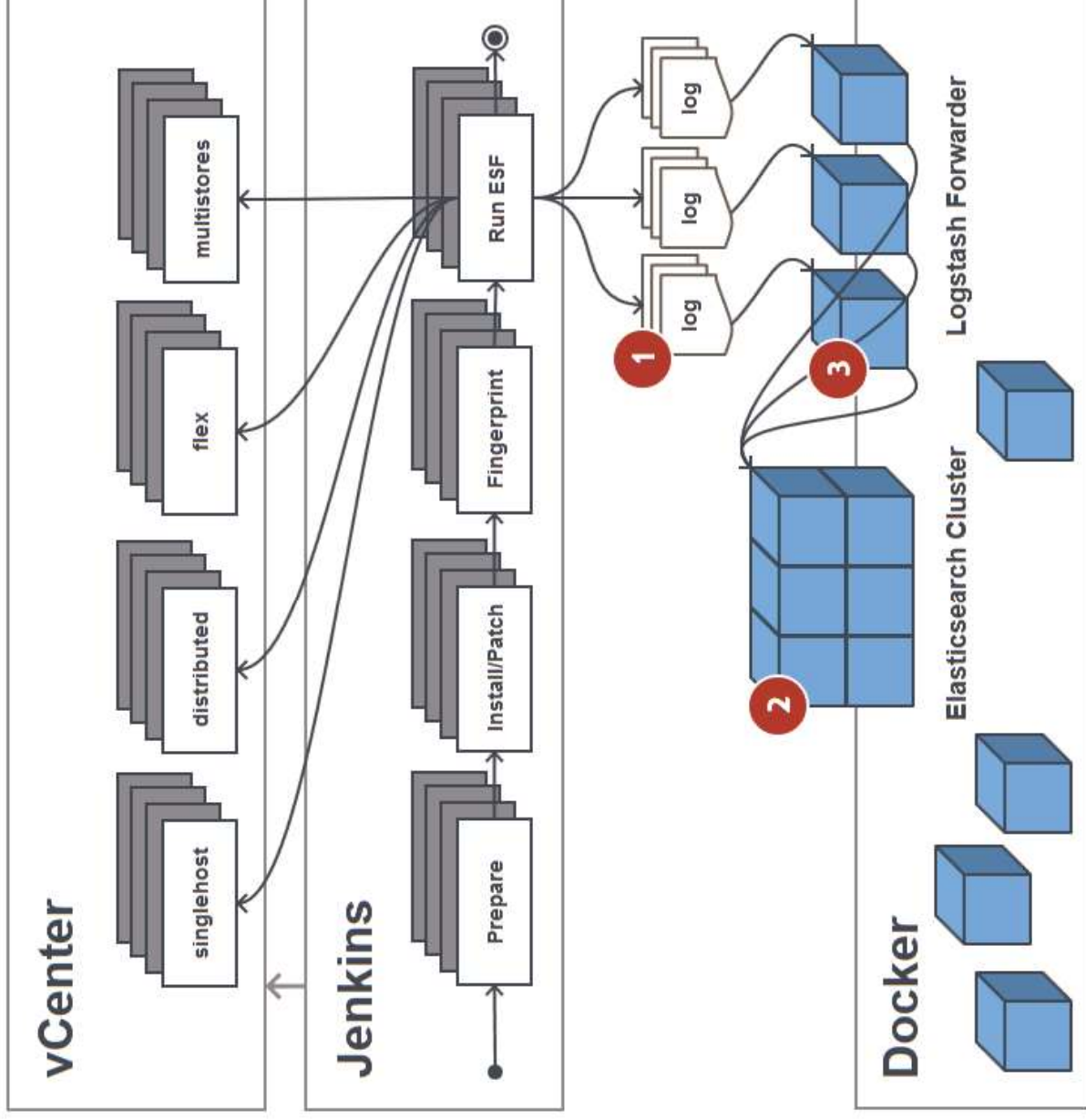
if [[ -z "${ES_IMAGE_TAG}" ]] ; then
    echo 'Variable ES_IMAGE_NAME is not set.'
    exit 1
fi

# We pull the official Image and only if that doesn't work are we
# triggering the build step.
which docker > /dev/null 2>&1
if [[ $? -ne 0 ]] ; then echo 'Docker is not installed.' ; fi
docker pull ${ES_IMAGE_NAME}:${ES_IMAGE_TAG}
if [[ $? -ne 0 ]] ; then
    echo 'Pulling Image was not successful. Triggering local Image build...'
    ${SCRIPT_DIR}/build.sh || exit 1
fi

# Stop running instance.
${SCRIPT_DIR}/stop.sh
```

Part #3: Logstash

Test Aggregation Workflow



to-logstash/config/logstash-esf.conf

```
input {  
  # Read esf log as events  
  # Wrap events as message in JSON object  
}  
  
filter {  
  # Process/transform/enrich events  
}  
  
output {  
  # Log to console  
  # Ship events to elasticsearch  
  # and index them as documents  
  # Write info/debug/error log  
}
```

to-logstash/config/logstash-esf.conf

```
input {
  {#- only if esf log should be processed #}
  {%- if "log" in LS_INPUT %}
    #####
    # Read esf log #
    #####
    # read from files via pattern
    file {
      path => [ "{{ LS_LOG_VOL }}" / "{{ LS_PATTERN }}" ]
      start_position => "beginning"
    }
    {%- endif %}
  }
}
```

to-logstash/config/logstash-esf.conf

```
filter {
  {#- only if esf log should be processed #}
  {%- if "log" in LS_INPUT %}

  # exclude empty and whitespace lines
  if [message] != "" and [message] !~ /^[\s]*$/ {

  #####
  # Add source fields in desired order #
  #####

  # only if no error tags were created
  if (![tags]) {

    # add needed env variables to event
    mutate {
      add_field => {
        "note" => ""
        "epages_version" => "{{ EPAGES_VERSION }}"
        "epages_repo_id" => "{{ EPAGES_REPO_ID }}"
        "env_os" => "{{ ENV_OS }}"
        "env_identifier" => "{{ ENV_IDENTIFIER }}"
        "env_type" => "{{ ENV_TYPE }}"
      }
    }
  }

  # extract esf fields from message; the content wrapper
  json { source => "message" }

  ...
}
```

to-logstash/config/logstash-est.conf

```
filter {
...
  # only if no error tags were created
  if (![tags]) {

    # add needed env variables to event
    mutate {
      add_field => {
        "report_url" => "{{ ENV_URL }}%{test_url}"
      }
    }

#####
# Remove not needed source fields #
#####

  # only if no error tags were created
  if (![tags]) {

    # remove not needed fields from extraction of message
    mutate { remove_field => [ "host", "message", "path",
      "test_url", "@timestamp", "@version" ] }

  }

...
}
```

to-logstash/config/logstash-esf.conf

```
filter {
...
#####
# Create document id #
#####

  if [env_identifier] != "zdt" {
    # generate document logstash id from several esf fields
    fingerprint {
      target => "[@metadata][ES_DOCUMENT_ID]"
      source => ["epages_repo_id", "env_os", "env_type",
                "env_identifier", "browser", "class", "method"]

      concatenate_sources => true
      key => "any-long-encryption-key"
      method => "SHA1" # return the same hash if all values of source fields
    }
  } else {
    # do not overwrite results for zdt environment identifier
    fingerprint {
      target => "[@metadata][ES_DOCUMENT_ID]"
      source => ["epages_repo_id", "env_os", "env_type",
                "env_identifier", "browser", "class", "method", "report_url"]

      concatenate_sources => true
      key => "any-long-encryption-key"
      method => "SHA1" # return the same hash if all values of source fields
    }
  } # end exclude whitespace
  {%- endif %}
}
}
```

to-logstash/config/logstash-esf.conf

```
output {  
  {%- if "verbose" in LS_OUTPUT or "console" in LS_OUTPUT %}  
    #####  
    # Output for verbose or console #  
    #####  
    # print all esf events as pretty json (info and error)  
    stdout { codec => rubydebug { metadata => true } }  
  {%- endif %}  
  ...  
}
```

to-logstash/config/logstash-est.conf

```
output {
... {%- if "elasticsearch" in LS_OUTPUT or "document" in LS_OUTPUT or "template" in LS_OUTPUT

#####
# Output for elasticsearch #
#####

    elasticsearch {

        hosts => {{ ES_HOSTS }}

        {%- if ES_USER and ES_PASSWORD %}
        user => "{{ ES_USER }}"
        password => "{{ ES_PASSWORD }}"
        {%- endif %}

        {%- if "elasticsearch" in LS_OUTPUT or "document" in LS_OUTPUT %}
        index => "{{ ES_INDEX }}"
        document_type => "{{ ES_DOCUMENT_TYPE }}"
        document_id => "%{[@metadata][ES_DOCUMENT_ID]}"
        {%- endif %}

        {%- if "elasticsearch" in LS_OUTPUT or "template" in LS_OUTPUT %}
        manage_template => true
        template => "{{ LS_CONFIG_VOL }}/template-est.json"
        template_name => "{{ ES_INDEX }}"
        template_overwrite => true
        {%- endif %}
    }
    {%- endif %}
...
}
```


to-logstash/config/logstash-esf.conf

```
output {
...
  {%- if "log" in LS_OUTPUT or "info" in LS_OUTPUT %}
  #####
  # Output for info log #
  #####
  # only if no error tags were created
  if (![tags]) {
    # log esf events to logstash output data
    file {
      path => "{{ LS_LOG_VOL }}/{{ LS_INFO }}"
      codec => "json" # cannot be changed
    }
  }
  {%- endif %}
  {%- if "log" in LS_OUTPUT or "error" in LS_OUTPUT %}
  #####
  # Output for error log #
  #####
  # if error tags were created during input processing
  if [tags] {
    # log failed esf events to logstash filter errors
    file {
      path => "{{ LS_LOG_VOL }}/{{ LS_ERROR }}"
      codec => "json" # cannot be changed
    }
  }
  {%- endif %}
}
```

to-logstash/circle.yml

```
machine:
  pre:
    # Configure elasticsearch circle service.
    - sudo cp -v "/home/ubuntu/to-logstash/test/service-elasticsearch.yml" "/etc/elasticsearch/hosts:"
  hosts:
    elasticsearch.circleci.com: 127.0.0.1
  services:
    - elasticsearch
    - docker
  environment:
    # Circle run tests with parallelism.
    CIRCLE_PARALLEL: true
    # Tests use dedicated docker containers, log directories and elasticsearch indexes.
    TEST_SAMPLE: "to-logstash-test-process-sample"
    TEST_PRODUCTION: "to-logstash-test-deploy-production"
    ... # SET ENV VARS

dependencies:
  override:
    ... # CONFIGURE DOCKER
    # Make sure circle project parallelism is set to at least 2 nodes.
    - |
      if [[ "${CIRCLE_NODE_TOTAL}" -eq "1" ]]; then {
        echo "Parallelism [${CIRCLE_NODE_TOTAL}x] needs to be 2x to fasten execution time."
        echo "You also need to set our circle env CIRCLE_PARALLEL [${CIRCLE_PARALLEL}] to tr
      }; fi

test:
  ...
```

to-logstash/circle.yml

```
test:
  override:
    - ? >
      case $CIRCLE_NODE_INDEX in
        0)
          printf "\n%s\n" "+++ Begin test of docker container [${TEST_SAMPLE}] +++"
          printf "\n%s\n\n" "=== Prepare test and setup config and log dirs on host ==="
          export LS_DOCKER_CONTAINER="${TEST_SAMPLE}"
          export LS_LOG="/tmp/${TEST_SAMPLE}/log"
          export LS_CONFIG="/tmp/${TEST_SAMPLE}/config"
          export ES_INDEX="${TEST_SAMPLE}"
          mkdir -v -p ${LS_LOG} ${LS_CONFIG}
          cp -v -r config/* ${LS_CONFIG}/
          cp -v test/${TEST_LOG} ${LS_LOG}/
          printf "\n%s\n" "--- Prepare test completed."
          # Fire up the container
          ./start.sh; [[ $? -eq 1 ]] && exit 1
          # Sleep is currently needed as file input is handled as a data stream
          # see: https://github.com/logstash-plugins/logstash-input-file/issues/52
          sleep 50;
          # Stop the container.
          ./stop.sh; [[ $? -eq 1 ]] && exit 1
          # Test metrics from files including input, output and errors.
          ./test/test-metrics-from-files.sh; [[ $? -eq 1 ]] && exit 1
          # Test metrics from elasticsearch including input, template and documents.
          ./test/test-metrics-from-elasticsearch.sh; [[ $? -eq 1 ]] && exit 1
          printf "\n%s\n" "+++ End test of docker container [${TEST_SAMPLE}] +++"
          # Exit case statement if run in parallel else proceed to next case.
          $CIRCLE_PARALLEL && exit 0
        ;&
      1)
```

to-logstash/circle.yml

```
test:
  override:
    - ? >
      case $CIRCLE_NODE_INDEX in
        0)
          ...
        1)
          printf "\n%s\n" "+++ Begin test of [${TEST_PRODUCTION}] +++"
          printf "\n%s\n" "=== Prepare test and setup config and log dirs on host ==="
          export LS_DOCKER_CONTAINER="${TEST_PRODUCTION}"
          export LS_LOG="/tmp/${TEST_PRODUCTION}/log"
          export LS_CONFIG="/tmp/${TEST_PRODUCTION}/config"
          export ES_INDEX="${TEST_PRODUCTION}"
          mkdir -v -p ${LS_LOG} ${LS_CONFIG}
          cp -v -r config/* ${LS_CONFIG}/
          cp -v test/${TEST_LOG} ${LS_LOG}/
          printf "\n%s\n" "--- Prepare test completed."
          # Run the full deploy script as used in jenkins.
          ./deploy.sh; [[ $? -eq 1 ]] && exit 1
          # Test metrics from files including input, output and errors.
          ./test/test-metrics-from-files.sh; [[ $? -eq 1 ]] && exit 1
          # Test metrics from elasticsearch including input, template and documents.
          ./test/test-metrics-from-elasticsearch.sh; [[ $? -eq 1 ]] && exit 1
          printf "\n%s\n" "+++ End test of [${TEST_PRODUCTION}] +++"
          # Exit case statement if run in parallel else proceed to next case.
          $CIRCLE_PARALLEL && exit 0
          ;&
        esac
      : parallel: true
    post:
      ...
```

to-logstash/circle.yml

```
test:
  override:
    - ? >
      case $CIRCLE_NODE_INDEX in
        0) ...
        1) ...
      esac
      : parallel: true
  post:
    - ? >
      case $CIRCLE_NODE_INDEX in
        0)
          printf "\n%s\n" "==== Archive artifacts of [${TEST_SAMPLE}] ===="
          sudo mv -v -f "/tmp/${TEST_SAMPLE}" "${CIRCLE_ARTIFACTS}/"
          mkdir -v -p "${CIRCLE_ARTIFACTS}/${TEST_SAMPLE}/services"
          sudo cp -v "${ES_CONF}" "${ES_LOG}" "$_"
          # Exit case statement if run in parallel else proceed to next case.
          $CIRCLE_PARALLEL && exit 0
          ;&
        1)
          printf "\n%s\n" "==== Archive artifacts of [${TEST_PRODUCTION}] ===="
          sudo mv -v -f "/tmp/${TEST_PRODUCTION}" "${CIRCLE_ARTIFACTS}/"
          mkdir -v -p "${CIRCLE_ARTIFACTS}/${TEST_PRODUCTION}/services"
          sudo cp -v "${ES_CONF}" "${ES_LOG}" "$_"
          # Exit case statement if run in parallel else proceed to next case.
          $CIRCLE_PARALLEL && exit 0
          ;&
      esac
      : parallel: true
```

to-logstash/circle.yml

```
general:  
  artifacts:  
    - "${CIRCLE_ARTIFACTS}/${LS_DOCKER_TEST_SAMPLE}"  
    - "${CIRCLE_ARTIFACTS}/${LS_DOCKER_TEST_PRODUCTION}"
```

.../test-metrics-from-files.sh

```
#!/bin/bash
# Test metrics of logstash files: LS_ERRORS_FILE, LS_INPUT_FILE, LS_OUTPUT_FILE.

# Set flag for exit error.
EXIT_ERROR=0

# Path to input, output and errors.
[[ "${LS_LOG}" ]] || { echo "ERROR: LS_LOG is not set"; exit 1; }
[[ "${TEST_LOG}" ]] && LS_INPUT_PATH="${LS_LOG}/${TEST_LOG}" || { echo "ERROR: TEST_LOG is not
[[ "${LS_INFO}" ]] && LS_OUTPUT_PATH="${LS_LOG}/${LS_INFO}" || { echo "ERROR: LS_INFO is not
[[ "${LS_ERROR}" ]] && LS_ERROR_PATH="${LS_LOG}/${LS_ERROR}" || { echo "ERROR: LS_ERROR is not

#####
# Files #
#####

# The input file with esf test results should exist.
printf "\n%s\n" "==== Find logstash input ===";
test -f ${LS_INPUT_PATH} && { printf "\n%s\n\n" "--- Following input log found: ${LS_INPUT_PAT

# The info log with logstash events should exist.
printf "\n%s\n" "==== Find logstash output ===";
test -f ${LS_OUTPUT_PATH} && { printf "\n%s\n\n" "--- Following info log found: ${LS_OUTPUT_PA

# The errors file with incorrectly transformed logstash events should not exist.
printf "\n%s\n" "==== Find logstash errors ===";
test -e ${LS_ERROR_PATH} && { printf "\n%s\n\n" "--- Following error log found: ${LS_ERROR_PAT

...

```

.../test-metrics-from-files.sh

```
...
#####
# Metrics #
#####

# The esf test results are transformed to logstash events.
# The esf test results are enriched with jenkins env variables.

# Collect metrics.
printf "\n%s\n" "==== Test metrics from log files ===="
LS_INPUT_LINES=`wc --lines < ${LS_INPUT_PATH}`
LS_INPUT_LENGTH=`wc --max-line-length < ${LS_INPUT_PATH}`
LS_OUTPUT_LINES=`wc --lines < ${LS_OUTPUT_PATH}`
LS_OUTPUT_LENGTH=`wc --max-line-length < ${LS_OUTPUT_PATH}`

# Print metrics.
printf "\n%s\n" "--- Count of lines from input log (${LS_INPUT_LINES}) and output log (${LS_OUT
printf "\n%s\n" "--- Maximum length from input log (${LS_INPUT_LENGTH}) should be less than ou

# Test metrics.
test "${LS_INPUT_LINES}" -eq "${LS_OUTPUT_LINES}" || EXIT_ERROR=1
test "${LS_INPUT_LENGTH}" -lt "${LS_OUTPUT_LENGTH}" || EXIT_ERROR=1

# Use exit error flag.
exit "${EXIT_ERROR}"
```


.../test-metrics-from-elasticsearch.sh

```
#!/bin/bash
# Test metrics of elasticsearch: ES_TEMPLATE and documents at ES_INDEX.

# Set flag for exit error.
EXIT_ERROR=0

# Print info.
printf "\n%s\n" "==== Test logstash data metrics from elasticsearch ${ES_HOSTS} ===="

# Path to elasticsearch template and logstash input test sample.
[[ "${LS_CONFIG}" ]] || { echo "ERROR: LS_CONFIG is not set"; exit 1; }
[[ "${ES_TEMPLATE}" ]] && ES_TEMPLATE_PATH="${LS_CONFIG}/${ES_TEMPLATE}" \
  || { [[ $LS_OUTPUT == *"elasticsearch"* || $LS_OUTPUT == *"template"* ]] \
    && { echo "ERROR: ES_TEMPLATE is not set"; exit 1; }
  || "${LS_LOG}" ]] || { echo "ERROR: LS_LOG is not set"; exit 1; }
[[ "${TEST_LOG}" ]] && LS_INPUT_PATH="${LS_LOG}/${TEST_LOG}" \
  || { [[ $LS_OUTPUT == *"elasticsearch"* || $LS_OUTPUT == *"documents"* ]] \
    && { echo "ERROR: TEST_LOG is not set"; exit 1; } }

# Check elasticsearch connection details.
[[ "${ES_HOSTS}" ]] || { echo "ERROR: ES_HOSTS is not set"; exit 1; }
[[ "${ES_USER}" ]] || { echo -e "\nINFO: ES_USER is optional and not set. Check if hosts ${ES_
[[ "${ES_PASSWORD}" ]] || { echo -e "\nINFO: ES_PASSWORD is optional and not set. Check if hos

# Set ES_HOSTS as hosts array.
declare -a HOSTS=`echo ${ES_HOSTS} | tr '[]' '(' | tr ', ' ' '`

...

```

.../test-metrics-from-elasticsearch.sh

```
..
#####
# Documents #
#####

# Fetch documents from all hosts.
[[ $LS_OUTPUT == *"elasticsearch"* || $LS_OUTPUT == *"documents"* ]] && {
    printf "\n%s\n" "==== Fetch documents from elasticsearch index [${ES_INDEX}] ===="
    ES_DOCUMENT_COUNTER=0
    for host in "${HOSTS[@]}";
    do
        printf "\n%s\n\n" "--- Following document count fetched: ${host}/${ES_INDEX}"
        ES_DOCUMENT_COUNTER=$((ES_DOCUMENT_COUNTER \
            + `curl --silent -u ${ES_USER}:${ES_PASSWORD} -XGET "${host}/${ES_INDEX}/_count?pretty" \
                | grep -E '.*count.*' | grep -E -o '[0-9]{1,}'`)
    done

    # Collect metrics.
    printf "%s\n" "==== Test metrics for elasticsearch documents ===="
    LS_INPUT_COUNT_LINES=`wc -l < ${LS_INPUT_PATH}`
    ES_DOCUMENT_COUNT_AVG=`expr ${ES_DOCUMENT_COUNTER} / ${#HOSTS[@]}`

    # Print metrics.
    printf "\n%s\n" "--- Count of lines from input log (${LS_INPUT_COUNT_LINES}) and average doc

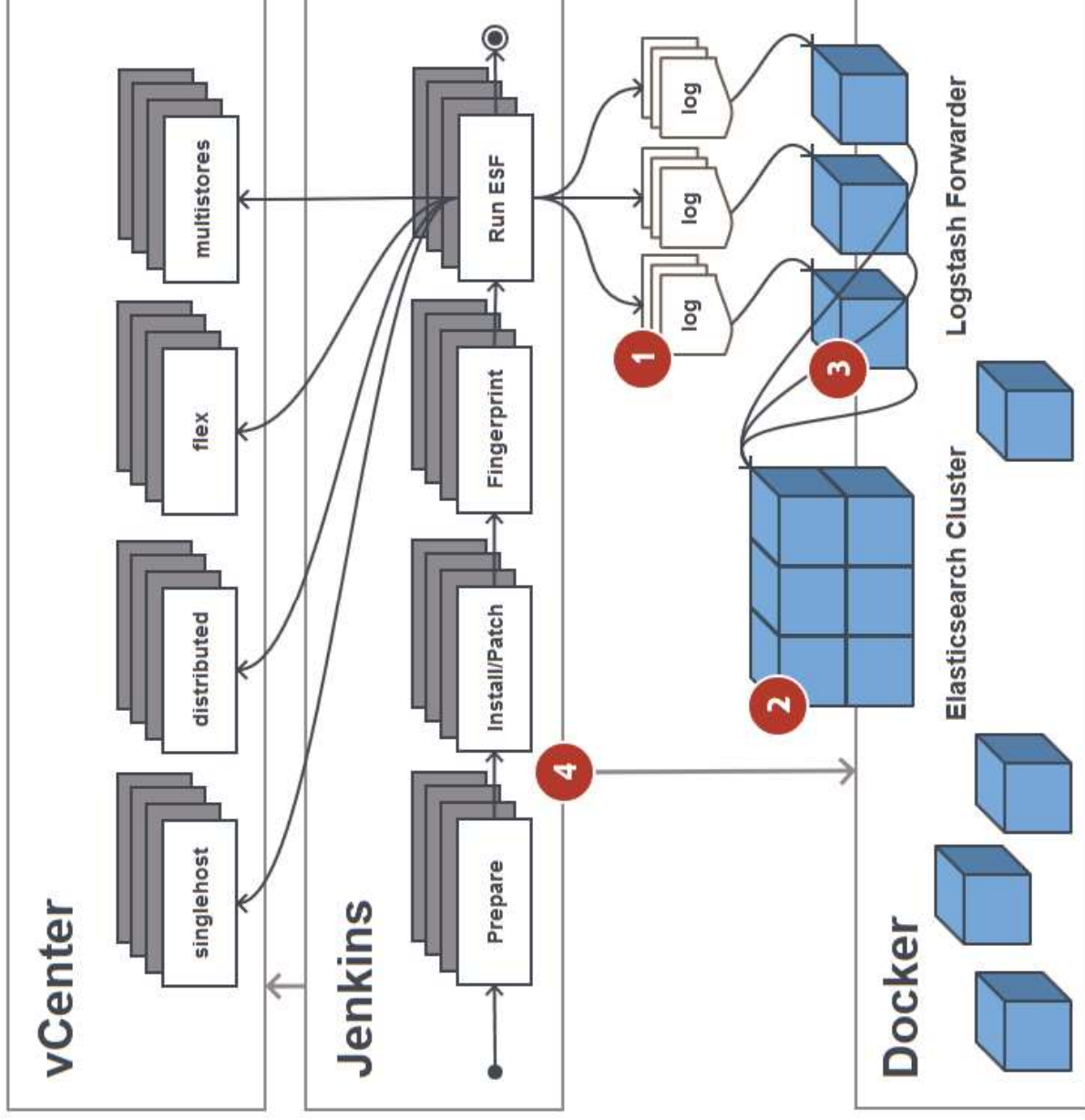
    # Test metrics.
    test "${LS_INPUT_COUNT_LINES}" -eq "${ES_DOCUMENT_COUNT_AVG}" || EXIT_ERROR=1
}

# Use exit error flag.
exit "${EXIT_ERROR}"
```



Part #4 Integration

Test Aggregation Workflow



Jenkins - Deploy - Elasticsearch

Setup:

- Checkout repo from Github
- Set ES_DATA, ES_LOGS

Build Steps:

```
#!/bin/bash

# Run puppet
/usr/bin/puppet agent --test
if [[ $? -eq 2 ]]; then exit 0 ; fi

# Create mounted directories
if [[ -n "${ES_DATA}" ]] && ! -d "${ES_DATA}" ]] ; then
    echo "Creating data directory ${ES_DATA} for Elasticsearch..."
    mkdir -p "${ES_DATA}"
fi
if [[ -n "${ES_LOGS}" ]] && ! -d "${ES_LOGS}" ]] ; then
    echo "Creating log directory ${ES_LOGS} for Elasticsearch..."
    mkdir -p "${ES_LOGS}"
fi

# Run deploy script for elasticsearch cluster
export BUILD_ID=dontKillMe
/jenkins/git/to-elasticsearch/deploy.sh
```

Jenkins - Run_ESF and forward logs

```
#!/bin/bash
export DISPLAY=":0"

SARGUMENT=
if [[ "${STORE}" ]]; then SARGUMENT="--store-name ${STORE}" ; fi
SDARGUMENT=
if [[ "${SHOP_DOMAIN}" ]]; then SDARGUMENT="--shop-domain ${SHOP_DOMAIN}" ; fi
SUARGUMENT=
if [[ "${SITE_URL}" ]]; then SUARGUMENT="--site-url http://${SITE_URL}/epages" ; fi
SSLPARGUMENT=
if [[ "${SSL_PROXY}" ]]; then SSLPARGUMENT="--ssl-proxy ${SSL_PROXY}" ; fi
WSARGUMENT=
if [[ "${WSADMIN_PASSWORD}" ]]; then WSARGUMENT="--soap-system-password ${WSADMIN_PASSWORD}"
RARGUMENT=
if [[ "${RETRY_TESTS}" == 'true' ]]; then RARGUMENT='--retry' ; fi
QARGUMENT=
if [[ "${RUN_QUARANTINE_TESTS}" ]]; then QARGUMENT="--quarantine" ; fi
SKIPARGUMENT=
if [[ "${SKIPPRECONDITIONS}" ]]; then SKIPARGUMENT="-ap 0 -sp" ; fi

if [[ -x bin/esf-epages6 ]]; then
    echo "bin/esf-epages6 -browser firefox -groups ${TESTGROUPS} --restart-browser -shop ${SARGUMENT} bin/esf-epages6 --language en_GB -browser firefox -groups ${TESTGROUPS} --restart-browser
    -url http://${TARGET_DOMAIN}/epages -email max.mustermann@epages.com --csv-report log/ee
    ${SARGUMENT} ${SDARGUMENT} ${SUARGUMENT} ${SSLPARGUMENT} ${WSARGUMENT} ${QARGUMENT} ${SKIPARGUMENT} ${SUARGUMENT} ${SDARGUMENT} ${SUARGUMENT} ${SSLPARGUMENT} ${WSARGUMENT} ${QARGUMENT} ${SKIPARGUMENT} ${SUARGUMENT}
    EXIT_CODE_ESF="?"
else
    exit 1
fi
...
```

Jenkins - Run_ESF and forward logs

```
if [[ $VERSION && $REPO && $ENV_TYPE && $ENV_IDENTIFIER && $ENV_OS ]]; then
    # push the esf-test-results.json to our elasticsearch server via logstash docker container

    # mount dirs
    export LS_LOG="$(find ${WORKSPACE} -mindepth 3 -maxdepth 3 -name "log" -type d)"
    export LS_CONFIG="${WORKSPACE}/to-logstash/config"

    # logstash.conf
    export LS_INPUT="log,esf"
    export LS_OUTPUT="log,elasticsearch"

    # pages6
    export EPAGES_VERSION=${VERSION}
    export EPAGES_REPO_ID=${REPO}

    # env url to dir ".../esf/.../log"
    export ENV_URL="${BUILD_URL}artifact/esf/${LS_LOG#*/esf/}"

    # elasticsearch connection details
    export ES_HOSTS="[ 'host.de:9200' ]"
    export ES_USER
    export ES_PASSWORD

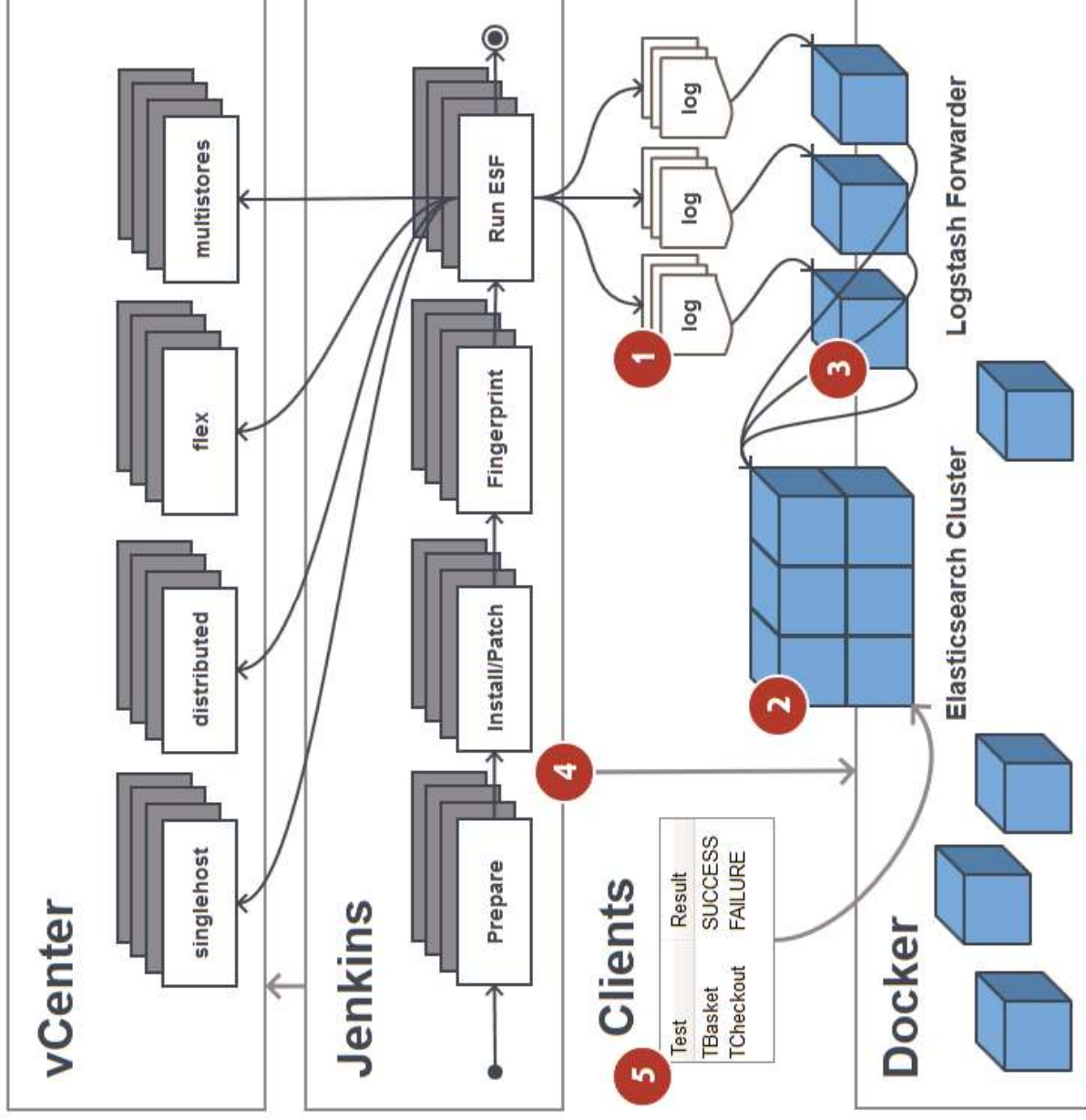
    # elasticsearch document path
    export ES_INDEX="esf-cdp-ui-tests"
    export LS_DOCKER_CONTAINER="to-logstash-run-esf-tests-${BUILD_NUMBER}"
    ${WORKSPACE}/to-logstash/deploy.sh || EXIT_CODE_LOGSTASH=1
    sudo chown -R jenkins:jenkins "${WORKSPACE}" || EXIT_CODE_LOGSTASH=1
fi

if [[ ${EXIT_CODE_ESF} -ne 0 || ${EXIT_CODE_LOGSTASH} -ne 0 ]]; then exit 1 ; fi
```



Part #5: Clients

Test Aggregation Workflow



Client 1: Static Page ESF Results for each pipeline run

Release Pipeline ESF Results

Lucene Query: (result::FAILURE or result::SKIP) AND NOT env_identifier:"quarantine" AND opages_repo_id:"6.17.48/2016.06.10-12.21.12"

| team | env_os | env_type | test | result | runtime | report_url |
|---------|--------|----------|---|---------|---------|----------------------|
| ePages6 | debian | patch | SuggestSearchSFTest.suggestSearchTest | FAILURE | 33 | link |
| ePages6 | debian | patch | AddCategoryTest.addCategoryViaNewItemAndAssignProduct | FAILURE | 97 | link |
| ePages6 | debian | patch | NewProductTest.createProductTest | FAILURE | 65 | link |
| ePages6 | debian | patch | NewBundleProductTest.createBundleProductTest | FAILURE | 43 | link |
| Ocean | centos | install | DdtIntrashipTest.testSetupAndCheckout | FAILURE | 304 | link |

Client 2: Elasticsearch Client via Lucene Request Query

Configuration
Search
Mapping
Cluster

esf-cdp-ui-tests

6.17.39

6 selected

Routing key:

Lucene Query:

.

result

Desc Asc

Delete By Id

Delete By Query

Search Index

Total results found: [4493] in 4 ms

Show 10 entries

| env_identifier | env_os | env_type | epages_version | result |
|----------------|--------|----------|----------------|---------|
| zdt | debian | patch | 6.17.39 | FAILURE |
| zdt | debian | patch | 6.17.39 | FAILURE |
| singlehost | centos | install | 6.17.39 | FAILURE |
| multistores | centos | install | 6.17.39 | FAILURE |
| zdt | debian | patch | 6.17.39 | FAILURE |
| singlehost | centos | install | 6.17.39 | FAILURE |
| singlehost | centos | install | 6.17.39 | FAILURE |
| singlehost | centos | install | 6.17.39 | FAILURE |
| multistores | centos | install | 6.17.39 | FAILURE |
| zdt | debian | patch | 6.17.39 | FAILURE |
| singlehost | centos | install | 6.17.39 | FAILURE |
| singlehost | centos | install | 6.17.39 | FAILURE |
| singlehost | centos | install | 6.17.39 | FAILURE |
| singlehost | centos | install | 6.17.39 | SKIP |
| singlehost | centos | install | 6.17.39 | SKIP |

JSON data

```

note : Unable to log in to Docdata. After ESF build
this will disappear.
epages_version : 6.17.39,
epages_repo_id : 6.17.39/2016.01.18-15.25.58,
env_os : centos,
env_identifier : singlehost,
env_type : install,
browser : firefox,
timestamp : 2016.01.18-19.10.16,
pos : 2,
result : FAILURE,
test : DocdataTest.testPaymentSetup,
class :
com.epages.cartridges.de_epages.docdata.tests.Docd
method : testPaymentSetup,
runtime : 139,
stacktrace :
org.openqa.selenium.TimeoutException: Timed out
after 30 seconds waiting for presence of element
located by: By.id: 'dropmenudiv Build info: version:
'2.47.1', revision: 'unknown', time: '2015-07-30
                    
```

Showing 1 to 20 of 50 entries
Ok Edit

Client 3: Angular App via DropDown Menu

Report

Quarantine

Version: 6.17.48

Search ePages Repo ids

6.17.48/2016.06.10-12.21.12

| Result | Klasse | Methode | Identifier | OS | URL | Team |
|---------|---|--|-------------------------|--------|-------------|---------|
| SUCCESS | com.epages.cartridges.de_epages.content.tests.AddCategoryTest | addCategoryViaContentPreviewAndAssignProduct | distributed_three_hosts | centos | 192.168.1.1 | ePages6 |
| SUCCESS | com.epages.cartridges.de_epages.content.tests.AddCategoryTest | addCategoryViaContentPreviewAndAssignProduct | flex_store | centos | 192.168.1.1 | ePages6 |
| SUCCESS | com.epages.cartridges.de_epages.content.tests.AddCategoryTest | addCategoryViaContentPreviewAndAssignProduct | multistores | centos | 192.168.1.1 | ePages6 |
| SUCCESS | com.epages.cartridges.de_epages.content.tests.AddCategoryTest | addCategoryViaContentPreviewAndAssignProduct | singlehost | centos | 192.168.1.1 | ePages6 |
| SUCCESS | com.epages.cartridges.de_epages.content.tests.AddCategoryTest | addCategoryViaContentPreviewAndAssignProduct | singlehost | centos | 192.168.1.1 | ePages6 |
| SUCCESS | com.epages.cartridges.de_epages.content.tests.AddCategoryTest | addCategoryViaContentPreviewAndAssignProduct | singlehost | debian | 192.168.1.1 | ePages6 |
| FAILURE | com.epages.cartridges.de_epages.content.tests.AddCategoryTest | addCategoryViaNewMenuItemAndAssignProduct | zdt | debian | 192.168.1.1 | ePages6 |
| SUCCESS | com.epages.cartridges.de_epages.content.tests.AddCategoryTest | addCategoryViaNewMenuItemAndAssignProduct | distributed_three_hosts | centos | 192.168.1.1 | ePages6 |
| SUCCESS | com.epages.cartridges.de_epages.content.tests.AddCategoryTest | addCategoryViaNewMenuItemAndAssignProduct | flex_store | centos | 192.168.1.1 | ePages6 |
| SUCCESS | com.epages.cartridges.de_epages.content.tests.AddCategoryTest | addCategoryViaNewMenuItemAndAssignProduct | multistores | centos | 192.168.1.1 | ePages6 |
| SUCCESS | com.epages.cartridges.de_epages.content.tests.AddCategoryTest | addCategoryViaNewMenuItemAndAssignProduct | singlehost | centos | 192.168.1.1 | ePages6 |
| SUCCESS | com.epages.cartridges.de_epages.content.tests.AddCategoryTest | addCategoryViaNewMenuItemAndAssignProduct | singlehost | centos | 192.168.1.1 | ePages6 |
| SUCCESS | com.epages.cartridges.de_epages.content.tests.AddCategoryTest | addCategoryViaNewMenuItemAndAssignProduct | singlehost | debian | 192.168.1.1 | ePages6 |
| SUCCESS | com.epages.cartridges.de_epages.content.tests.AddCategoryTest | addCategoryViaNewMenuItemAndAssignProduct | zdt | debian | 192.168.1.1 | ePages6 |
| SUCCESS | com.epages.cartridges.de_epages.content.tests.AddCategoryTest | addCategoryViaNewMenuItemAndAssignProduct | zdt | debian | 192.168.1.1 | ePages6 |
| SUCCESS | com.epages.cartridges.de_epages.shop.tests.CurrencyTest | addCurrencyTest | distributed_three_hosts | centos | 192.168.1.1 | ePages6 |
| SUCCESS | com.epages.cartridges.de_epages.shop.tests.CurrencyTest | addCurrencyTest | flex_store | centos | 192.168.1.1 | ePages6 |
| SUCCESS | com.epages.cartridges.de_epages.shop.tests.CurrencyTest | addCurrencyTest | multistores | centos | 192.168.1.1 | ePages6 |
| SUCCESS | com.epages.cartridges.de_epages.shop.tests.CurrencyTest | addCurrencyTest | singlehost | centos | 192.168.1.1 | ePages6 |
| SUCCESS | com.epages.cartridges.de_epages.shop.tests.CurrencyTest | addCurrencyTest | singlehost | centos | 192.168.1.1 | ePages6 |
| SUCCESS | com.epages.cartridges.de_epages.shop.tests.CurrencyTest | addCurrencyTest | singlehost | debian | 192.168.1.1 | ePages6 |

Retrospective



Result
Where did we succeed?
Where did we succeed?



Result Where did we mess up?

What does it mean for epages?

- Abolish Release & Test Automation Team
- COP Integration/Testing/Languages

What does it mean for us?

- Microservices Architecture
- Become "real" DevOps with Automation Skills

checklist

epages

Where to start?

People & Processes

- 1) Analyze your issue!
- 2) Find common sense on technical implementation
- 3) Use implementation techniques

Learnings





Common Mistakes

to-elasticsearch/start.sh

```
#!/bin/bash

# Exit 1 if any script fails.
set -e

#####
# elasticsearch #
#####

# The ES_CONFIG, ES_DATA and ES_LOGS dirs are mandatory and mounted to DOCKER_CONFIG_VOL, DOCF
[[ -z "${ES_CONFIG}" ]] && { echo "ERROR: Elasticsearch config directory ${ES_CONFIG} is not set
[[ ! -d "${ES_CONFIG}" ]] && { echo "Elasticsearch config directory ${ES_CONFIG} does not exist
HOST_CONFIG_DIR=${ES_CONFIG}"
[[ -z "${ES_DATA}" ]] && { echo "ERROR: Elasticsearch data directory ${ES_DATA} is not set."; ex
[[ ! -d "${ES_DATA}" ]] && { echo "Elasticsearch data directory ${ES_DATA} does not exist."; e
HOST_DATA_DIR=${ES_DATA}"
[[ -z "${ES_LOGS}" ]] && { echo "ERROR: Elasticsearch logs directory ${ES_LOGS} is not set."; ex
[[ ! -d "${ES_LOGS}" ]] && { echo "Elasticsearch logs directory ${ES_LOGS} does not exist."; e
HOST_LOGS_DIR=${ES_LOGS}"

# The ES_ENV file is mandatory and defines our used settings. Some settings can be set at runt
[[ -z "${ES_ENV}" ]] && { echo "ERROR: ES_ENV is not set."; exit 1; }
[[ -z "${ES_HEAP_SIZE}" ]] && { export ES_HEAP_SIZE="1g"; echo -e "\nINFO: ES_HEAP_SIZE is opt

# Map ES host ports.
HOST_HTTP_PORT="9200"; [[ -n "${ES_HOST_HTTP}" ]] && HOST_HTTP_PORT=${ES_HOST_HTTP}
HOST_TCP_PORT="9300"; [[ -n "${ES_HOST_TCP}" ]] && HOST_TCP_PORT=${ES_HOST_TCP}

#####
# docker #
#####
```

to-logstash/start.sh

```
#!/bin/bash

# Exit 1 if any script fails.
set -e

#####
# logstash #
#####

# The LS_LOG and LS_CONFIG dirs are mandatory and mounted to DOCKER_LOG_VOL and DOCKER_CONFIG_
[[ -z "${LS_LOG}" ]] && { echo "ERROR: Logstash log directory LS_LOG is not set."; exit 1; };
[[ ! -d "${LS_LOG}" ]] && { echo "ERROR: Logstash log directory [LS_LOG=${LS_LOG}] does not ex
HOST_LOG_DIR=${LS_LOG}";
[[ -z "${LS_CONFIG}" ]] && { echo "ERROR: Logstash config directory LS_CONFIG is not set."; ex
[[ ! -d "${LS_CONFIG}" ]] && { echo "ERROR: Logstash config directory [LS_CONFIG=${LS_CONFIG}]
HOST_CONFIG_DIR=${LS_CONFIG}";

# The LS_INPUT type is mandatory and sets our logstash input.
[[ "${LS_INPUT}" =~ ^.*(conf)|(log)|(esf).*$ ]] || {
    echo "ERROR: Logstash input [LS_INPUT=${LS_INPUT}] is not set correctly. Possible values: [c

# The LS_OUTPUT types is mandatory and set our logstash output.
[[ "${LS_OUTPUT}" =~ ^.*(conf)|(verbose)|(console)|(log)|(info)|(error)|(elasticsearch)|(docur
    echo "ERROR: Logstash output [LS_OUTPUT=${LS_OUTPUT}] is not set correctly. Possible values:

# The LS_INFO and LS_ERROR log files have default names.
[[ -z "${LS_INFO}" ]] && export LS_INFO="logstash-info.json";
[[ -z "${LS_ERROR}" ]] && export LS_ERROR="logstash-error.json";

#####
# esf config #
.....
```

to-logstash/docker-entrypoint.sh

```
#!/bin/bash

# Exit 1 if any script fails.
set -e

# Add logstash as command if needed
if [[ "${1:0:1}" = '-' ]]; then
    set -- logstash "$@"
fi

# If running logstash
if [[ "$1" = 'logstash' ]]; then

    # Change the ownership of the mounted volumes to user logstash at docker container runtime
    chown -R logstash:${LS_CONFIG_VOL} ${LS_LOG_VOL}

    # Get verbose and log settings
    [[ "${LS_OUTPUT}" =~ ^.*(verbose).*$ ]] && VERBOSE=true || VERBOSE=false
    [[ "${LS_OUTPUT}" =~ ^.*(log)|(info)|(error).*$ ]] && LOG=true || LOG=false

    # Get LS_ENV
    {
        # Find env vars in docker
        $VERBOSE && printf "\n%s\n" "=== Find env vars defined in docker container from list [
        LS_ENV_PATH=$( find "${LS_CONFIG_VOL}" -maxdepth 3 -iname "${LS_ENV}" )
        $VERBOSE && {
            [[ -f "${LS_ENV_PATH}" ]] \
                && { printf "\n%s\n\n" "---- Following env list found: ${LS_ENV_PATH}"; cat ${I
                || printf "\n%s\n\n" "---- No env list file found."
        }
        $VERBOSE && {
```




Sources

Docker - Best Practices

- <http://crosbymichael.com/dockerfile-best-practices.html>
- <http://crosbymichael.com/dockerfile-best-practices-take-2.html>
- <https://getcarina.com/docs/best-practices/docker-best-practices-dockerfile/>
- <http://www.carlboettiger.info/2014/08/29/docker-notes.html>
- <https://www.digitalocean.com/community/tutorials/docker-explained-using-dockerfiles-to-automate-building-of-images>
- <http://jonathan.bergknoff.com/journal/building-good-docker-images>
- <http://www.infoq.com/interviews/lindsay-docker>

Logstash

- Based on | [Logstash Dockerfile](#) from Official Docker Library
- Reference | [Current Docs](#)

Logstash configuration plugins

- Input | [file](#)
- Filter | [json](#)
- Filter | [mutate](#)
- Filter | [environment](#)
- Filter | [fingerprint](#)
- Output | [stdout](#)
- Output | [elasticsearch](#)
- Output | [file](#)

Elasticsearch

- [Reference](#) | [Configuration](#)
- [Reference](#) | [Module HTTP \(9200\)](#)
- [Reference](#) | [Module TCP \(9300\)](#)
- [Reference](#) | [Module Slowlog](#)
- [Reference](#) | [Plugins](#)
- [Plugin](#) | [head](#)
- [Plugin](#) | [http-basic](#)
- [Client](#) | [ESClient](#)

Dockerfile

- [Based on](#) | [Elasticsearch Dockerfile](#) at [Official Docker Library](#)
- [Ideas from](#) | [Elasticsearch Dockerfile](#) at [Official Docker Trusted](#)
- [Ideas from](#) | [Elasticsearch Dockerfile](#) at [Official CircleCI Examples](#)
- [Ideas from](#) | [Java Dockerfile](#) at [Official Docker Library](#)

Elasticsearch index template

- [Reference | Indices API > Index Templates](#)
- [Reference | Indices API > Mappings](#)
- [Reference | Indices API > Aliases](#)
- [Blog | Aliases upon index creation](#)

July Questions

Follow us



[dataduke](#)

[dastianoro](#)

[epagesdevs](#)



developer.epages.com/blog