DevOps Fundamentals Course Handout

Would you like to know more? This handout and bibliography provides definitions of terms used in the course as well as links to the topics, tools and resources we reference.

**Index:**

[Introduction](#_aac7o2p29xvn)

[Chapter 1: DevOps Basics](#_jlqig8q3g1i9)

[What Is DevOps?](#_5r6doh6wlu16)

[DevOps Core Values: CAMS](#_l5gb7pifxizq)

[DevOps Core Values: The Three Ways](#_cff2ywzhh2n0)

[Your DevOps Playbook](#_bjzd9h1xwt3r)

[Ten Practices for DevOps Success: 10-6](#_9bq0mdss5vnq)

[Ten Practices for DevOps Success: 5-1](#_xg82cu3qlmpk)

[DevOps Tools - The Cart Or The Horse?](#_lppozdyz19ug)

[Chapter 2: DevOps: A Culture Problem](#_s1ot4z1cjl8i)

[The IT Crowd and the Coming Storm](#_oqxi2p58lnm)

[Use Your Words](#_y9e8kx3qdzon)

[Do Unto Others](#_kpd7si1zap5p)

[Throwing Things Over Walls](#_v32kh1fhd98p)

[Kaizen: Continuous Improvement](#_io9hwy8icb)

[Chapter 3: The Building Blocks of DevOps](#_ai5l323skkfm)

[DevOps Building Block: Agile](#_mydafa3jgqvw)

[DevOps Building Block: Lean](#_o7gkd57d7l3y)

[ITIL, ITSM, and the SDLC](#_kxn7yzd7d18o)

[Chapter 4: Infrastructure Automation](#_g4amwtx6a90d)

[Infrastructure As Code](#_oi5ka3xqydsa)

[Golden Image to Foil Ball](#_lutodzoo0q73)

[Immutable Deployment](#_b3z6u39orjzl)

[Your Infrastructure Toolchain](#_y8w16548levz)

[Chapter 5: Continuous Delivery](#_q6kkw5grm6v4)

[Small + Fast = Better](#_f7gk8mgn8ocz)

[Continuous Integration Practices](#_pxfzadwq0lel)

[The Continuous Delivery Pipeline](#_huha6dqu91d8)

[The Role Of QA](#_k8wvuk18uv2o)

[Your CI Toolchain](#_dc7uf656nr8v)

[Chapter 6: Reliability Engineering](#_wyjhw8k337kk)

[Engineering Doesn't End With Deployment](#_iurvrn3uxiyy)

[Design For Operation - Theory](#_j3727ie4y8jd)

[Design For Operation - Practice](#_jy6ju5sauu80)

[Operate For Design: Metrics and Monitoring](#_8q1hltir2ccd)

[Operate for Design: Logging](#_8g5o4bt9s8i5)

[Your SRE Toolchain](#_trwjilo6vcxr)

[Chapter 7: Additional DevOps Resources](#_8g4wvwsrm15w)

[Unicorns, Horses, and Donkeys, Oh My](#_l3sy3b85rahq)

[Ten Best DevOps Books You Need to Read](#_jbsd3zg0rle)

[Navigating The Series of Tubes](#_n32apdo6w1lx)

[Chapter 8: The Future of DevOps](#_pnsazqib3r3q)

[Cloud to Containers to Serverless](#_uctslvzfa750)

[The Rugged Frontier of DevOps: Security](#_5ufu2upugzhe)

[Chapter 9: Conclusion](#_1whont4e9p08)

[Next Steps: Am I a DevOp now?](#_aj1njng1fm3y)

## Introduction

Your fearless instructors are:

James Wickett (@wickett) and Ernest Mueller (@ernestmueller)

<https://theagileadmin.com/>

# 

# 

# Chapter 1: DevOps Basics

## What Is DevOps?

**DevOps** - is the practice of operations and development engineers participating together through the entire service lifecycle; from the design and development process all the way to production support. DevOps is also characterized by operations staff making use of many of the same techniques as developers for their systems work. <https://theagileadmin.com/what-is-devops/>

2015 State of DevOps Report

<https://puppet.com/resources/white-paper/2015-state-devops-report>

2016 State of DevOps Report

<https://puppet.com/resources/white-paper/2016-state-devops-report>

## DevOps Core Values: CAMS

**CAMS** - Culture, Automation, Measurement, Sharing

What DevOps Means To Me, by John Willis <https://www.chef.io/blog/2010/07/16/what-devops-means-to-me/>

DevOps Culture, by John Willis

<http://itrevolution.com/devops-culture-part-1/>

People over Process over Tools, by Damon Edwards <http://dev2ops.org/2010/02/people-over-process-over-tools/>

## DevOps Core Values: The Three Ways

**The Three Ways**

1. Systems Thinking
2. Amplifying Feedback Loops
3. A Culture of Continuous Experimentation and Learning

The Three Ways, by Gene Kim  
<http://itrevolution.com/the-three-ways-principles-underpinning-devops/>

## Your DevOps Playbook

**5 Key DevOps Methodologies**

1. People over Process over Tools
2. Continuous Delivery
3. Lean Management
4. Visible Ops style Change Control
5. Infrastructure as Code

People over Process over Tools, by Damon Edwards

<http://dev2ops.org/2010/02/people-over-process-over-tools/>

Continuous Delivery, by Jez Humble and David Farley <https://www.amazon.com/Continuous-Delivery-Deployment-Automation-Addison-Wesley/dp/0321601912>

2015 State of DevOps Report

<https://puppet.com/resources/white-paper/2015-state-devops-report>

2016 State of DevOps Report

<https://puppet.com/resources/white-paper/2016-state-devops-report>

The Amazing DevOps Transformation Of The HP LaserJet Firmware Team (Gary Gruver), by Gene Kim

<http://itrevolution.com/the-amazing-devops-transformation-of-the-hp-laserjet-firmware-team-gary-gruver/>

Leading the Transformation, by Gary Gruver and Tommy Mouser

<http://itrevolution.com/books/leading-the-transformation/>

The Visible Ops Handbook: Implementing ITIL in 4 Practical and Auditable Steps, by Gene Kim, Kevin Behr, and George Spafford

<https://www.amazon.com/Visible-Ops-Handbook-Implementing-Practical/dp/0975568612>

## Ten Practices for DevOps Success: 10-6

**10 Practices for DevOps Success**

1. Embedded Teams
2. Blameless Postmortems
3. Status Pages
4. Developers On Call
5. Incident Command System

Incident Command for IT: What We Can Learn From The Fire Department, by Brent Chapman

<https://www.usenix.org/legacy/event/lisa05/tech/chapman.pdf>

Keys to SRE, by Ben Treynor

<https://www.usenix.org/conference/srecon14/technical-sessions/presentation/keys-sre>

Transparent Uptime, by Lenny Rachitsky

<http://www.transparentuptime.com/>

How Complex Systems Fail, by Dr. Richard Cook

<http://web.mit.edu/2.75/resources/random/How%20Complex%20Systems%20Fail.pdf>

Blameless Postmortems, by John Allspaw

<https://codeascraft.com/2012/05/22/blameless-postmortems/>

## Ten Practices for DevOps Success: 5-1

**10 Practices for DevOps Success**

1. Chaos Monkey
2. Blue/Green Deployments
3. Dependency Injection
4. Andon Cords
5. The Cloud

Dependency Injection, by Martin Fowler

<http://martinfowler.com/articles/injection.html>

Chaos Monkey Released Into The Wild, by Cory Bennett and Ariel Tseitlin

<http://techblog.netflix.com/2012/07/chaos-monkey-released-into-wild.html>

The Andon Cord, by John Willis

<http://itrevolution.com/kata/>

## DevOps Tools - The Cart Or The Horse?

TL;DR: Both

DevOps Toolchain

<https://en.wikipedia.org/wiki/DevOps#DevOps_toolchain>

DevOps Borat

<https://twitter.com/DEVOPS_BORAT/status/41587168870797312>

# 

# 

# Chapter 2: DevOps: A Culture Problem

## The IT Crowd and the Coming Storm

The IT Crowd

<http://www.imdb.com/title/tt0487831/>

Dilbert, by Scott Adams

<http://dilbert.com/>

BOFH, by Simon Travaglia

<http://www.theregister.co.uk/data_centre/bofh/>

User Friendly, by J.D. Frazer

<http://www.userfriendly.org/>

Xkcd, by Randall Munroe

<http://xkcd.com/>

What Is DevOps, by Damon Edwards

<http://dev2ops.org/2010/02/what-is-devops/>

10+ Deploys Per Day: Dev and Ops Cooperation at Flickr, by John Allspaw and John Hammond

<http://www.slideshare.net/jallspaw/10-deploys-per-day-dev-and-ops-cooperation-at-flickr>

## Use Your Words

**Blameless Postmortems contain:**

1. A description of the incident
2. A description of the root cause
3. How the incident was stabilized or fixed.
4. A timeline of events including all actions taken to resolve the incident
5. How the incident affected customers
6. Remediations and corrective actions.

**Transparent Uptime means:**

1. Admit Failure
2. Sound Like A Human
3. Have A Communication Channel
4. Above All Else, Be Authentic

Blameless Postmortems, by John Allspaw

<https://codeascraft.com/2012/05/22/blameless-postmortems/>

A Guideline for Postmortem Communication, by Lenny Rachitsky

<http://www.transparentuptime.com/2010/03/guideline-for-postmortem-communication.html>

Rackspace’s Public Status Page

<https://status.rackspace.com/>

## Do Unto Others

**Trust Blockers:**

* Lack of Context
* Conflicting Goals

How To Win Friends and Influence People, by Dale Carnegie

<https://en.wikipedia.org/wiki/How_to_Win_Friends_and_Influence_People>

How To Say It At Work, by Jack Griffin

<https://www.amazon.com/How-Say-Work-Second-Communication/dp/0735204306>

The No A\*\*hole Rule, by Robert I. Sutton

<https://en.wikipedia.org/wiki/The_No_Asshole_Rule>

Getting To Yes, by Robert Fisher and William L. Ury

<https://en.wikipedia.org/wiki/Getting_to_Yes>

Crucial Conversations, by Kerry patterson, Joseph Grenny, Ron McMillan, and Al Switzler

<https://en.wikipedia.org/wiki/Crucial_Conversations:_Tools_for_Talking_When_Stakes_Are_High>

Hanlon’s Razor

<https://en.wikipedia.org/wiki/Hanlon%27s_razor>

Ringelmann Effect

<https://en.wikipedia.org/wiki/Ringelmann_effect>

Parkinson’s Law

<https://en.wikipedia.org/wiki/Parkinson%27s_law>

Dunbar’s Number

<https://en.wikipedia.org/wiki/Dunbar%27s_number>

Is Your Team Too Big? Too Small? What’s the Right Number?

<http://knowledge.wharton.upenn.edu/article/is-your-team-too-big-too-small-whats-the-right-number-2/>

Hubot, a chat bot

<https://hubot.github.com/>

Web Operations, by John Allspaw and Jesse Robbins

<https://www.amazon.com/Web-Operations-Keeping-Data-Time/dp/1449377440>

Effective DevOps, by Jennifer Davis and Katherine Daniels

<http://shop.oreilly.com/product/0636920039846.do>

## Throwing Things Over Walls

The Phoenix Project, by Gene Kim, Kevin Behr, George Spafford

<https://en.wikipedia.org/wiki/The_Phoenix_Project_(novel)>

DevOps Culture, by Martin Fowler

<http://martinfowler.com/bliki/DevOpsCulture.html>

Shadow IT

<https://en.wikipedia.org/wiki/Shadow_IT>

Conway’s Law

<https://en.wikipedia.org/wiki/Conway%27s_law>

Operations Maturity Model  
<https://pages.chef.io/operations-maturity-model>

A Typology of Organisational Cultures, by Ron Westrum

<http://www.ncbi.nlm.nih.gov/pmc/articles/PMC1765804/pdf/v013p0ii22.pdf>

## Kaizen: Continuous Improvement

**Kaizen** - change for the better

**Kaizen’s Guiding Principles**

* Good processes bring good results
* Go see for yourself to grasp the current situation (gemba)
* Speak with data, manage by facts
* Take action to contain and correct root causes of problems
* Work as a team
* Kaizen is everybody’s business

Kaizen Glossary

<https://us.kaizen.com/knowledge-center/glossary.html>

When In Japan…, by Ryan Day

<http://www.qualitydigest.com/inside/lean-article/100115-when-japan.html#>

Kaizen

<https://en.wikipedia.org/wiki/Kaizen>

Toyota Kata

<https://en.wikipedia.org/wiki/Toyota_Kata>

5 Whys

<https://en.wikipedia.org/wiki/5_Whys>

# Chapter 3: The Building Blocks of DevOps

## DevOps Building Block: Agile

Agile Systems Administration Google Group  
<https://groups.google.com/forum/#!forum/agile-system-administration>

Agile Infrastructure, by Andrew Clay Shafer

<http://www.slideshare.net/littleidea/agile-infrastructure-velocity-09>

DevOpsDays Ghent 2007

<http://www.devopsdays.org/events/2009-ghent/>

Opscamp Austin 2010

<https://theagileadmin.com/2010/02/05/opscamp-debrief/>

Agile Manifesto

<http://www.agilemanifesto.org/>

Agile Principles

<http://agilemanifesto.org/principles.html>

10th Annual State of Agile Survey <https://versionone.com/pdf/VersionOne-10th-Annual-State-of-Agile-Report.pdf>

A DevOps Manifesto, by Ernest Mueller

<https://theagileadmin.com/2010/10/15/a-devops-manifesto/>

## DevOps Building Block: Lean

**7 Principles of Lean Software Development**

• ELIMINATE WASTE  
• AMPLIFY LEARNING  
• DECIDE AS LATE AS POSSIBLE  
• DELIVER AS FAST AS POSSIBLE  
• EMPOWER THE TEAM  
• BUILD INTEGRITY IN  
• SEE THE WHOLE

**The Seven Wastes (Muda) of Lean Software**

Waste #1 - Partially Done Work  
Waste #2 - Extra Features  
Waste #3 - Relearning  
Waste #4 - Handoffs  
Waste #5 - Delays  
Waste #6 - Task Switching  
Waste #7 - Defects

**Build-Measure-Learn**

• BUILD – MINIMUM VIABLE PRODUCT

• MEASURE – THE OUTCOME AND INTERNAL METRICS

• LEARN – ABOUT YOUR PROBLEM AND YOUR SOLUTION

• REPEAT – GO DEEPER WHERE IT’S NEEDED

Lean Manufacturing

<https://en.wikipedia.org/wiki/Lean_manufacturing>

Lean Software Development: An Agile Toolkit, by Mary and Tom Poppendieck

<https://www.amazon.com/Lean-Software-Development-Agile-Toolkit/dp/0321150783>

Lean Startup, by Eric Ries

<https://en.wikipedia.org/wiki/Lean_startup>

Value Stream Mapping

<https://en.wikipedia.org/wiki/Value_stream_mapping>

DevOps Culture, by John Willis

<http://itrevolution.com/devops-culture-part-1/>

## ITIL, ITSM, and the SDLC

ITSM

<https://en.wikipedia.org/wiki/IT_service_management>

Opscamp Austin 2010

<http://www.johnmwillis.com/opscamp/opscamp-austin-roundup/>

ITIL

<https://en.wikipedia.org/wiki/ITIL>

The Visible Ops Handbook: Implementing ITIL in 4 Practical and Auditable Steps, by Gene Kim, Kevin Behr, and George Spafford

<https://www.amazon.com/Visible-Ops-Handbook-Implementing-Practical/dp/0975568612>

# Chapter 4: Infrastructure Automation

## Infrastructure As Code

Infrastructures.org

<http://www.infrastructures.org/>

Architectures for open and scalable clouds, by Randy Bias

<http://www.slideshare.net/randybias/architectures-for-open-and-scalable-clouds>

Infrastructure as Code, by Martin Fowler

<http://martinfowler.com/bliki/InfrastructureAsCode.html>

## Golden Image to Foil Ball

**Provisioning** is the process of making a server ready for operation, including hardware, OS, system services, network connectivity.

**Deployment** is the process of automatically deploying and upgrading applications on a server.

**Orchestration** is the act of performing coordinated operations across multiple systems.

**Configuration management** is an overarching term dealing with change control of system configuration after initial provision, but is often also applied to maintaining and upgrading application and application dependencies.

**Imperative** - also known as “procedural,” this is an approach where commands desired to produce a state are defined and executed.

**Declarative** - also known as “functional,” this is an approach where you define a desired state and the tool converges the existing system on the model.

**Idempotent** - the ability to execute the CM procedure repeatedly and end up in the same state each time.

**Self service** - is the ability for an end user to kick off one of these processes without having to go through other people.

Server Provisioning

<https://en.wikipedia.org/wiki/Provisioning#Server_provisioning>

Golden Image or Foil Ball, by Luke Kanies

<http://madstop.com/post/85950592485/golden-image-or-foil-ball-repost>

Canary Release, by Martin Fowler

<http://martinfowler.com/bliki/CanaryRelease.html>

Blue-Green Deployment, by Martin Fowler

<http://martinfowler.com/bliki/BlueGreenDeployment.html>

## Immutable Deployment

AMI Creation with Aminator, by Michael Tripoli & Karate Vick

<http://techblog.netflix.com/2013/03/ami-creation-with-aminator.html>

Immutable Server, by Martin Fowler

<http://martinfowler.com/bliki/ImmutableServer.html>

Immutable Delivery, by John Willis

<https://theagileadmin.com/2015/11/24/immutable-delivery/>

CMDB

<https://en.wikipedia.org/wiki/Configuration_management_database>

The CMDB is Dead, Long Live the CMDB, by Rhonabwy

<https://rhonabwy.com/2010/07/18/the-cmdb-is-dead-long-live-the-cmdb/>

Hadoop and Zookeeper

<http://hadoop.apache.org/>

Awesome Sysadmin tool list, by Francisco Augusto

<https://github.com/kahun/awesome-sysadmin>

Collins

<http://tumblr.github.io/collins/>

## Your Infrastructure Toolchain

AWS Cloudformation

<https://aws.amazon.com/cloudformation/>

Azure ARM Templates

<https://azure.microsoft.com/en-us/documentation/templates/>

Hashicorp Terraform

<https://www.terraform.io/>

Ubuntu Juju

<http://www.ubuntu.com/cloud/juju>

Chef

<https://www.chef.io/>

Puppet

<https://puppet.com/>

Ansible

<https://www.ansible.com/>

Saltstack

<https://saltstack.com/community/>

Cfengine

<https://cfengine.com/>

Rubocop

<http://batsov.com/rubocop/>

Foodcritic

<http://www.foodcritic.io/>

Chefspec

<https://docs.chef.io/chefspec.html>

kitchenCI

<http://kitchen.ci/>

Ohai

<https://docs.chef.io/ohai.html>

Etcd

<https://coreos.com/etcd/docs/latest/>

Zookeeper

<https://wiki.apache.org/hadoop/ZooKeeper>

Consul

<https://www.consul.io/>

Docker

<https://www.docker.com/>

Docker swarm

<https://docs.docker.com/swarm/>

Kubernetes

<http://kubernetes.io/>

Mesos

<http://mesos.apache.org/>

Rancher

<http://rancher.com/>

Google cloud platform container engine

<https://cloud.google.com/container-engine/docs/>

Amazon ECS

<https://aws.amazon.com/ecs/>

habitat

<https://www.habitat.sh/>

# Chapter 5: Continuous Delivery

## Small + Fast = Better

**Continuous Delivery has the following benefits:**

1. Time to market goes down
2. Quality increases not decreases
3. Limits your Work In Progress
4. Shortens lead times for changes
5. Improves Mean Time To Recover

Continuous Delivery, by Jez Humble and David Farley <https://www.amazon.com/Continuous-Delivery-Deployment-Automation-Addison-Wesley/dp/0321601912>

Deming's 14 Key Points

<https://www.deming.org/theman/theories/fourteenpoints>

## Continuous Integration Practices

**To successfully perform Continuous Integration:**

1. Builds should pass the coffee test (< 5 minutes)
2. Commit really small bits
3. Don’t leave the build broken
4. Use a trunk based development flow
5. Don't allow flaky tests, fix them!
6. The build should return a status, a log, and an artifact

## The Continuous Delivery Pipeline

**To successfully perform Continuous Delivery:**

1. Only build artifacts once
2. Artifacts should be immutable
3. Deployment should go to a copy of production before going into production
4. Stop deploys if it a previous step fails
5. Deployments should be idempotent

## The Role Of QA

**Types of Testing:**

* Unit Testing
* Code Hygiene
* Integration Testing
* TDD / BDD /ATDD
* Infrastructure Testing
* Performance Testing
* Security Testing

The Difference Between TDD, BDD, and ATDD

http://www.assertselenium.com/atdd/difference-between-tdd-bdd-atdd/

## Your CI Toolchain

**Version Control**

Git

<https://git-scm.com/>

Subversion

<https://subversion.apache.org/>

Github

<https://github.com/>

Bitbucket

<https://bitbucket.org/>

Perforce

<https://www.perforce.com/>

**Continuous Integration**

Jenkins

<https://jenkins.io/>

GoCD

<https://www.go.cd/>

Bamboo

<https://www.atlassian.com/software/bamboo>

TeamCity

<https://www.jetbrains.com/teamcity/>

Travis CI

<https://travis-ci.org/>

Circle CI

<https://circleci.com/>

**Build**

Make

<https://www.gnu.org/software/make/>

Rake

<https://github.com/ruby/rake>

Maven

<https://maven.apache.org/>

Gulp

<http://gulpjs.com/>

Packer

<https://www.packer.io/>

Fpm

<https://github.com/jordansissel/fpm/wiki>

**Test**

JUnit

<http://junit.org/junit4/>

Golint

<https://github.com/golang/lint>

Go fmt

<https://golang.org/cmd/gofmt/>

Rubocop

<http://batsov.com/rubocop/>

Findbugs

<http://findbugs.sourceforge.net/>

Robot Framework

<http://robotframework.org/>

Protractor

<http://www.protractortest.org/#/>

Cucumber

<https://cucumber.io/>

Selenium

<http://www.seleniumhq.org/>

Sauce Labs

<https://saucelabs.com/>

KitchenCI

<http://kitchen.ci/>

Apachebench

<https://httpd.apache.org/docs/2.4/programs/ab.html>

JMeter

<http://jmeter.apache.org/>

Gauntlt

<http://gauntlt.org/>

Mittn

<https://github.com/F-Secure/mittn>

**Artifact Repository**

Artifactory

<https://www.jfrog.com/artifactory/>

Nexus

<http://www.sonatype.org/nexus/>

Bintray

<https://bintray.com/>

Dockerhub

<https://hub.docker.com/>

Amazon S3

<https://aws.amazon.com/s3/>

**Deployment**

Rundeck

<http://rundeck.org/>

Urbancode

<https://developer.ibm.com/urbancode/products/urbancode-deploy/>

Thoughtworks

<https://www.thoughtworks.com/continuous-delivery>

Deployinator

<https://github.com/etsy/deployinator>

Tool Discovery Trick: Google one of these names “vs” and you’ll get autosuggestions of that tool versus other tools in its space other people are asking about!

# 

# 

# Chapter 6: Reliability Engineering

## Engineering Doesn't End With Deployment

Reliability Engineering

<https://en.wikipedia.org/wiki/Reliability_engineering>

2015 State of DevOps Report

<https://puppet.com/resources/white-paper/2015-state-devops-report>

2016 State of DevOps Report

<https://puppet.com/resources/white-paper/2016-state-devops-report>

Devops Areas - Codifying devops practices, by Patrick Debois

<http://www.jedi.be/blog/2012/05/12/codifying-devops-area-practices/>

Site Reliability Engineering, by Betsy Beyer, Chris Jones, Jennifer Petoff, Niall Richard Murphy

<http://shop.oreilly.com/product/0636920041528.do>

## 

## Design For Operation - Theory

Design Patterns, by Erich Gamma, Richard Helm, Ralph Johnson, John Vlissides

<https://www.amazon.com/Design-Patterns-Elements-Reusable-Object-Oriented/dp/0201633612>

Release It!, by Michael Nygard

<https://pragprog.com/book/mnee/release-it>

Hystrix

<https://github.com/Netflix/Hystrix>

The Twelve-Factor App

<http://12factor.net>

Martin Fowler’s Architecture Descriptions

<http://martinfowler.com/bliki/>

## Design For Operation - Practice

Netflix’ Chaos Monkey

<http://techblog.netflix.com/2012/07/chaos-monkey-released-into-wild.html>

Frequency Reduces Difficulty, by Martin Fowler

<http://martinfowler.com/bliki/FrequencyReducesDifficulty.html>

AWS outage: How Netflix weathered the storm by preparing for the worst

<http://www.techrepublic.com/article/aws-outage-how-netflix-weathered-the-storm-by-preparing-for-the-worst/>

Code profiling

<https://en.wikipedia.org/wiki/Profiling_(computer_programming)>

<https://en.wikipedia.org/wiki/List_of_performance_analysis_tools>

## Operate For Design: Metrics and Monitoring

**The 6 Monitoring Areas**

1. Service Performance and Uptime
2. Software Component Metrics
3. System Metrics
4. App Metrics
5. Performance
6. Security

How Complex Systems Fail, by Dr. Richard Cook

<http://web.mit.edu/2.75/resources/random/How%20Complex%20Systems%20Fail.pdf>

A Lean Cloud Monitoring Checklist, by Ernest Mueller

<http://slideplayer.com/slide/7650435/>

## Operate for Design: Logging

Logging and Log Management, by Anton Chuvakin, Kevin Schmidt, and Chris Phillips

<http://shop.oreilly.com/product/9781597496353.do>

## Your SRE Toolchain

Pingdom

<https://www.pingdom.com/>

Datadog

<https://www.datadoghq.com/>

Netuitive

<http://www.netuitive.com/>

Ruxit

<https://www.dynatrace.com/platform/offerings/ruxit/>

Librato

<https://www.librato.com/>

New Relic

<https://newrelic.com/>

AppDynamics

<https://www.appdynamics.com/>

Statsd

<https://github.com/etsy/statsd>

Ganglia

<http://ganglia.info/>

Graphite

<https://graphiteapp.org/>

Grafana

<http://grafana.org/>

InfluxDB

<https://influxdata.com/>

OpenTSDB

<http://opentsdb.net/>

Metrics

<http://metrics.dropwizard.io/>

Icinga

<https://www.icinga.org/>

Sensu

<https://sensuapp.org/>

Nagios

<https://www.nagios.org/>

Prometheus

<https://prometheus.io/>

Sysdig

<https://sysdig.com/>

Splunk

<http://www.splunk.com/>

SumoLogic

<https://www.sumologic.com/>

Logentries

<https://logentries.com/>

ELK Stack

<https://www.elastic.co/webinars/introduction-elk-stack>

PagerDuty

<https://www.pagerduty.com/>

VictorOps

<https://victorops.com/>

Flapjack

<http://flapjack.io/>

StatusPage

<https://www.statuspage.io/>

Rerun

<http://rerun.github.io/rerun/>

Rundeck

<http://rundeck.org/>

# 

# 

# Chapter 7: Additional DevOps Resources

## Unicorns, Horses, and Donkeys, Oh My

The DevOps Donkey Represents the Rest of Us

<http://devops.com/2015/05/05/the-devops-donkey-represents-the-rest-of-us/>

Gartner Says By 2016, DevOps Will Evolve From a Niche to a Mainstream Strategy Employed by 25 Percent of Global 2000 Organizations

<http://www.gartner.com/newsroom/id/2999017>

DevOpsDays

<http://www.devopsdays.org/>

Velocity Conference

<http://conferences.oreilly.com/velocity>

DevOps Enterprise Summit

<http://events.itrevolution.com/>

ChefConf

<https://chefconf.chef.io/>

AWS re:Invent

<https://reinvent.awsevents.com/>

Monitorama

<http://monitorama.com/>

Surge

<https://surge.omniti.com/2016>

Scale

<https://atscaleconference.com/>

Structure

<http://www.structureconf.com/>

DevOpsConferences.org

<http://devopsconferences.org>

Meetup.com

<https://www.meetup.com/>

## Ten Best DevOps Books You Need to Read

The Visible Ops Handbook: Implementing ITIL in 4 Practical and Auditable Steps, by Gene Kim, Kevin Behr, and George Spafford

<https://www.amazon.com/Visible-Ops-Handbook-Implementing-Practical/dp/0975568612>

Continuous Delivery, by Jez Humble and David Farley <https://www.amazon.com/Continuous-Delivery-Deployment-Automation-Addison-Wesley/dp/0321601912>

Release It!, by Michael Nygard

<https://pragprog.com/book/mnee/release-it>

Effective DevOps, by Jennifer Davis and Katherine Daniels

<http://shop.oreilly.com/product/0636920039846.do>

Lean Software Development: An Agile Toolkit, by Mary and Tom Poppendieck

<https://www.amazon.com/Lean-Software-Development-Agile-Toolkit/dp/0321150783>

Web Operations, by John Allspaw and Jesse Robbins

<https://www.amazon.com/Web-Operations-Keeping-Data-Time/dp/1449377440>

The Practice of Cloud System Administration, by Christine Hogan, Strata R. Chalup, and Thomas A. Limoncelli

<http://the-cloud-book.com/>

The DevOps Handbook, by Gene Kim, Jez Humble, Patrick Debois, and John Willis

<http://itrevolution.com/devops-handbook>

Leading the Transformation, by Gary Gruver and Tommy Mouser

<http://itrevolution.com/books/leading-the-transformation/>

The Phoenix Project, by Gene Kim, Kevin Behr, George Spafford

<https://en.wikipedia.org/wiki/The_Phoenix_Project_(novel)>

## Navigating The Series of Tubes

DevOps Weekly, by Gareth Rushgrove

<http://www.devopsweekly.com/>

@garethr

DevOps Google Group

<https://groups.google.com/forum/#!forum/devops>

DevOps Toolchain Google Group

<https://groups.google.com/forum/#!forum/devops-toolchain>

DevOps.com, Chief Editor: Alan Shimel

<http://devops.com/>

@devopsdotcom

@ashimmy

DZone

<https://dzone.com/devops-tutorials-tools-news>

InfoQ

<https://www.infoq.com/devops/>

DevOps Cafe, by John Willis and Damon Edwards

<http://devopscafe.org/>

@botchgalupe

@damonedwards

The Ship Show, by J. Paul Reed, Sascha Bates, Pete Cheslock, and Youssuf El-Kalay

<http://theshipshow.com/>

@jpaulreed

@sascha\_d

@petecheslock

@buildscientist

Arrested DevOps, by Matt Stratton, Trevor Hess, and Bridget Kromhout.

<https://www.arresteddevops.com/>

@arresteddevops

@bridgetkromhout

@mattstratton

@trevorghess

Food Fight, by Nathen Harvey

<http://foodfightshow.org/>

@foodfightshow

@nathenharvey

DevOps Mastery, by Brian Wagner

<http://www.devopsmastery.com/podcasts/>

@devopsmaster

Kitchen Soap, by John Allspaw

<http://www.kitchensoap.com/>

@allspaw

IT Revolution Press, by Gene Kim (and others)

<http://itrevolution.com/devops-blog/>

@RealGeneKim

Dev2Ops, by Damon Edwards and Alex Honor

<http://dev2ops.org/>

@damonedwards

@alexhonor

Continuous Delivery, by Jez Humble

<https://continuousdelivery.com/blog/>

@jezhumble

Stochastic Resonance, by Andrew Clay Shafer

<https://stochasticresonance.wordpress.com/>

@littleidea

JEDI, by Patrick Debois

<http://www.jedi.be/blog/>

@patrickdebois

More than Seven, by Gareth Rushgrove

<http://www.morethanseven.net/>

@garethr

The Agile Admin, by Ernest Mueller, James Wickett, and Karthik Gaekwad

<https://theagileadmin.com/>

@ernestmueller

@wickett

@iteration1

SlideShare

<http://www.slideshare.net/search/slideshow?searchfrom=header&q=devops>

YouTube

<https://www.youtube.com/results?search_query=devops>

Vimeo

<https://vimeo.com/search?q=devops>

# 

# Chapter 8: The Future of DevOps

## Cloud to Containers to Serverless

“**cloud computing** is a model for enabling ubiquitous, convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction”

NIST Special Publication 800-145

<http://www.nist.gov/itl/csd/cloud-102511.cfm>

RightScale 2016 State of the Cloud Survey

<https://www.rightscale.com/lp/state-of-the-cloud>

Docker Hub

<https://hub.docker.com/>

“**Serverless**… is run in stateless compute containers that are event-triggered, ephemeral (may only last for one invocation), and fully managed by a 3rd party.”

Serverless, by Martin Fowler

<http://martinfowler.com/articles/serverless.html>

AWS Lambda

<https://aws.amazon.com/lambda/>

Zappa

<https://zappa.gun.io/>

8 Surprising Facts About Real Docker Adoption

<https://www.datadoghq.com/docker-adoption/>

## The Rugged Frontier of DevOps: Security

The Rugged Manifesto

<https://www.ruggedsoftware.org/>

The DevOps Audit Defense Kit

<http://itrevolution.com/devops-and-auditors-the-devops-audit-defense-toolkit/>

Chef Audit and Compliance Framework: Inspec

<https://www.chef.io/inspec/>

ThreatStack

<https://www.threatstack.com/>

Gauntlt

<http://gauntlt.org/>

Signal Sciences

<https://www.signalsciences.com>

AlienVault

<https://alienvault.com>

# 

# 

# Chapter 9: Conclusion

## Next Steps: Am I a DevOp now?

Amazon Web Services

<https://aws.amazon.com/>

Docker

<https://docs.docker.com/engine/installation/>

Terraform

<https://www.terraform.io/>

<https://hub.docker.com/r/hashicorp/terraform/>

Jenkins

<https://jenkins.io/>

<https://hub.docker.com/_/jenkins/>

Github

<https://try.github.io/>

James Wickett (@wickett) and Ernest Mueller (@ernestmueller)

<https://theagileadmin.com/>