

DevOps Foundations: Infrastructure as Code Course Handout

Table of Contents

- [Glossary](#)
- [Bibliography](#)
- [Tools](#)

Glossary

Term	Definition
artifact	A set of deployable elements coming out of a build process
cloud computing	A method of managing computing resources characterized by on-demand, self-service processing, broad network access, resource pooling, rapid elasticity, and measured service
continuous delivery	The practice of deploying every CI build to a production-like environment and performing automated integration and acceptance testing after it passes its build and unit tests to ensure your code is always ready to deploy to production
continuous deployment (CD)	The practice of automatically pushing your code out when tests pass your continuous delivery pipeline
continuous integration (CI)	The practice of automatically building and unit testing an entire application frequently, ideally on every source code check-in, dozens of times a day, if necessary
configuration management (CM)	Change control of system configuration during and after initial provisioning, including maintaining and upgrading applications and application dependencies
container	A unit of software that allows you to run segmented minisystems on a host system, usually assigned only the necessary components required for a specific application to run. Containers only virtualize the layers above the host machine OS. Containers are intended to be immutable.
convergent	CM automation that runs repeatedly to converge systems on a desired state. Many of the declarative CM tools like Chef and Puppet are convergent.

declarative	An approach to configuration management developers in which you specify a desired state and let the configuration management tool determine how best to achieve it (also sometimes referred to as “functional”)
deployment	Putting a new application, or version of an existing application, on the system where it is intended to run
end-to-end testing	Tests actual user facing behavior of entire system
ephemeral	A compute resource intended to be dynamically created and destroyed frequently
gitops	A deployment methodology using declarative assets stored versioned and immutable in source control (usually git), pulled automatically from the repo and continuously reconciled with the desired system state by a tool or agent
idempotent	The ability to execute a CM procedure repeatedly and end up in the same state. Declarative frameworks are often idempotent and imperative ones often are not.
immutable	Created compute assets that are not intended to be changed after initial deployment, but instead redeployed if needed
imperative	An approach to configuration management in which developers define an order of actions to be taken in order to achieve the desired result (also sometimes referred to as “procedural”)
infrastructure as code (IaC)	Provisioning infrastructure and performing configuration management on systems by executing code instead of performing manual work. This approach can reduce system inconsistencies and enable testing early in development.
integration testing	Testing several components at once; for IaC testing, this tests creation and modification of system components to inspect them for fitness
Kubernetes	An open-source container orchestration system for automating software deployment, scaling, and management
managed service	A higher-level set of functionalities a cloud provider operates for you as a resource, like a cloud database or other complex subsystem
orchestration	The act of performing coordinated operations across multiple systems

provisioning	The process of making a server or other underlying infrastructure ready for operation, including hardware, OS, system services, and network connectivity
runbook	A runbook is an automated script to perform an administrative task inside your system
security testing	Testing that searches for positive and negative security attributes
serverless	A compute model where the underlying infrastructure is managed by a cloud provider or platform, and interaction is limited to supplying code to execute (also sometimes referred to as “Functions as a Service,” or FaaS)
unit testing	The smallest testable components tested in isolation
virtual machine	A technology that segments a server down to the hardware layer to create multiple running systems. More isolated but slower to provision and operate than containers

Bibliography

Infrastructure as Code: Dynamic Systems for the Cloud Age, by Kief Morris

DevOps Foundations course:

<https://www.lynda.com/Operating-Systems-tutorials/DevOps-Fundamentals/508618-2.html>

“What Is DevOps?”

<https://theagileadmin.com/what-is-devops/>

Martin Fowler on Infrastructure as Code:

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

Martin Fowler on The Practical Test Pyramid:

<https://martinfowler.com/articles/practical-test-pyramid.html>

Martin Fowler on The Strangler Fig Application:

<https://www.martinfowler.com/bliki/StranglerApplication.html>

Martin Fowler on Immutable Servers:

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

Git tutorial:

<https://try.github.io/levels/1/challenges/1>

Git Essential Training: The Basics (2019) course:

<https://www.linkedin.com/learning/git-essential-training-the-basics-2019/>

NIST Cloud Definition:

<http://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-145.pdf>

“Production Begins in Development”:

<http://www.slideshare.net/dev2ops/velocity-2011-production-begins-in-development>

Luke Kanies on “Golden Image or Foil Ball?”:

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

Immutable Delivery:

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

Google’s DevOps Research & Assessment Program:

<https://www.devops-research.com/research.html>

Tools

- AWS CloudFormation <https://aws.amazon.com/cloudformation/> - AWS infrastructure provisioning
- Azure ARM <https://azure.microsoft.com/en-us/get-started/azure-portal/resource-manager/> - Azure infrastructure provisioning
- Terraform <https://www.terraform.io/> - Cross cloud infrastructure provisioning (owned by Hashicorp)
 - Terraform Registry <https://registry.terraform.io/> - community Terraform modules
 - Terratest <https://terratest.gruntwork.io/> - Terraform test framework
 - Kitchen-terraform <https://github.com/newcontext-oss/kitchen-terraform> - Terraform test framework
 - Managing resource drift <https://developer.hashicorp.com/terraform/tutorials/state/resource-drift>
- Digital Rebar <https://rackn.com/products/rebar/> - Hardware friendly infrastructure provisioning
- DryRun Security <https://dryrun.security> - security testing for developers
- Boto3 <https://aws.amazon.com/sdk-for-python/> - the AWS SDK for Python
- Puppet <https://www.puppet.com/> - Configuration management system (now owned by Perforce)
- Chef <https://www.chef.io/> - Configuration management system (now owned by Progress)
- Ansible <https://www.ansible.com/> - Automation and config management framework (now owned by Red Hat)
 - Ansible Docs <https://docs.ansible.com/>
- Salt <https://saltproject.io/> - Automation and config management framework
- Serverspec <https://serverspec.org/> - An infrastructure testing framework
- Inspec <https://community.chef.io/tools/chef-inspec> - An infrastructure testing and compliance framework
- Rundeck <https://www.rundeck.com/> - Automation framework (now owned by PagerDuty)
- Packer <https://www.packer.io/> - System image builder (owned by Hashicorp)
- Git <https://git-scm.com/> - source code control system
 - Github <https://github.com/> - community code repositories (now owned by Microsoft)
- Make <https://www.gnu.org/software/make/> - a build execution tool
- Docker <https://www.docker.com/> - containerization framework
 - Docker Hub <https://hub.docker.com/> - community repo of container images

- Rkt <https://www.redhat.com/en/topics/containers/what-is-rkt> - Alternative container engine (now owned by Red Hat)
- Kubernetes <https://kubernetes.io/> - container orchestration framework
 - Cloud Native Computing Foundation <https://www.cncf.io/> - organization for k8s and all its thousands of related projects
 - Kubectl cheat sheet <https://www.blumatador.com/learn/kubectl-cheatsheet>
 - CRI-O <https://cri-o.io/> - lightweight container runner for k8s
 - Kubespray <https://kubespray.io/#/> - create a k8s cluster with terraform and ansible
- Helm <https://helm.sh/> - Kubernetes deployment tool
 - Artifact Hub <https://artifacthub.io/> - community repo of Kubernetes packages
- AWS Lambda <https://docs.aws.amazon.com/lambda/> - AWS' serverless engine
- WebAssembly (WASM) <https://webassembly.org/> - portable compilation target to ease app deployment
 - What is WASM? <https://wasmlabs.dev/articles/docker-without-containers/>
 - WASM + k8s: <https://www.cncf.io/blog/2022/11/17/better-together-a-kubernetes-and-wasm-case-study/>
- Nix <https://nixos.org/> - reproducible declarative build and artifact tool
 - “What Is Nix and Why You Should Use It” <https://serokell.io/blog/what-is-nix>
- System Initiative <https://www.systeminit.com/> - Adam Jacob, inventor of Chef, has a new cutting edge take on IaC going
- Serverless Framework <https://www.serverless.com/>
 - Free and commercial serverless framework
- Word Cloud Generator - <https://github.com/wickett/word-cloud-generator/>

Your friendly authors,

[Ernest Mueller](#) (@ernestmueller) and [James Wickett](#) (@wickett)