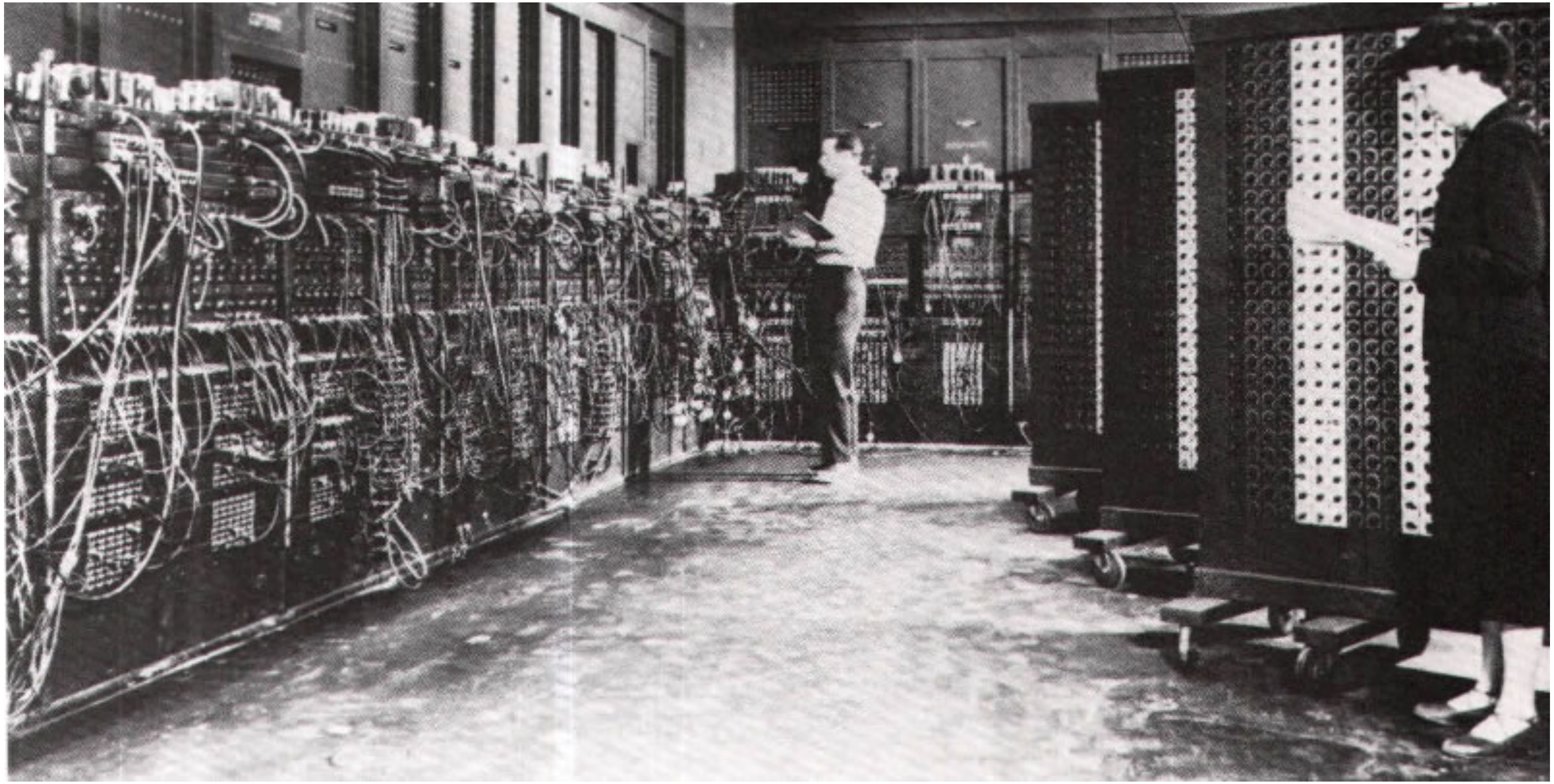
DEVSECOPS BOOTCAMP

BUILDING RUGGED SOFTWARE

YEAR ONE / WEEK TWO / LESSON ONE

How my Grandfather ran a stack…



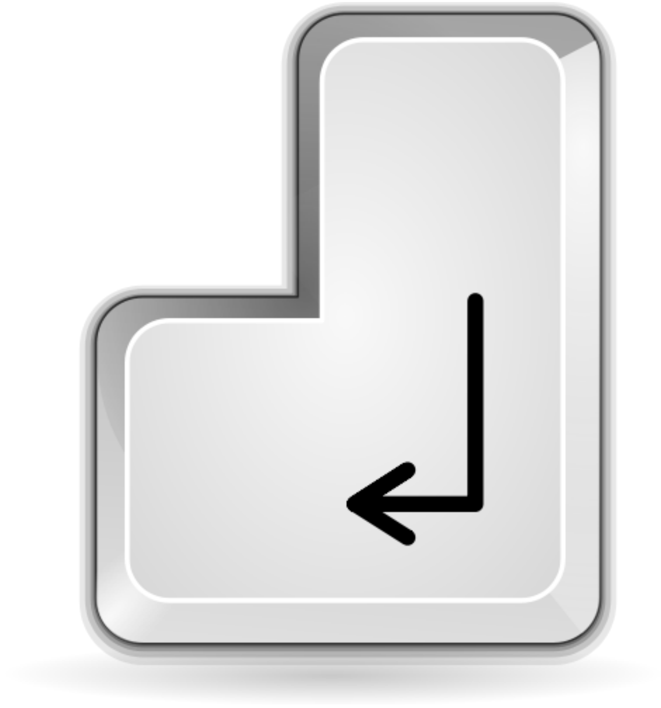
Glen Beck (background) and Betty Snyder (foreground) program ENIAC in BRL building 328. (U.S. Army photo)

How my Mother ran a stack…

Lawrence Livermore National Laboratory [Attribution], via Wikimedia Commons

How I run a stack…

```
ec2-run-instances ami-12345678 -t t1.micro -k my-key-pair -g my-security-group
```



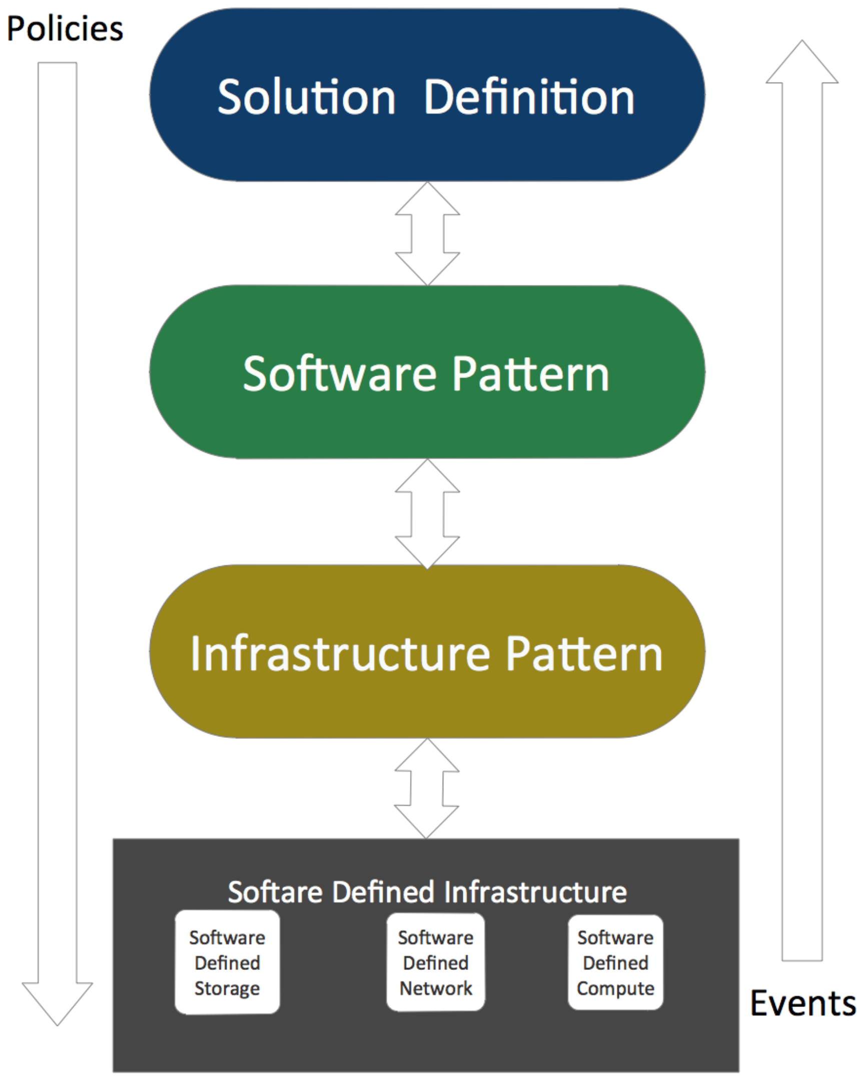
© 2007 Nuno Pinheiro & David Vignoni & David Miller & Johann Ollivier Lapeyre & Kenneth Wimer & Riccardo Iaconelli / KDE, via Wikimedia Commons

Software Defined Environment

- Virtualized/abstracted infrastructure managed by **software**, ie. Configuration as Code:
  - Chef
  - Puppet
  - AWS CloudFormation
- The software being deployed **defines** the configuration and virtualized infrastructure requirements.
- The virtualized **infrastructure** extends past the data center to allow for multiple environments.

Common Characteristics of SDEs

- Solution Definition
- Software Pattern
- Infrastructure Pattern
- Software Defined Infrastructure
  - Software Defined Network
  - Software Defined Storage
  - Software Defined Compute
  - Software Defined Security
  - Software Defined …



Benefits of SDEs

- Automatically adjusts to workload based on demand (autoscale)
- Centrally managed
- Everything as Code, underlying policies are code (JSON, YAML etc.)
- Better resource management
- Holistic overview of the environment
- Faster deployments
- Built-in audit trails and API endpoints

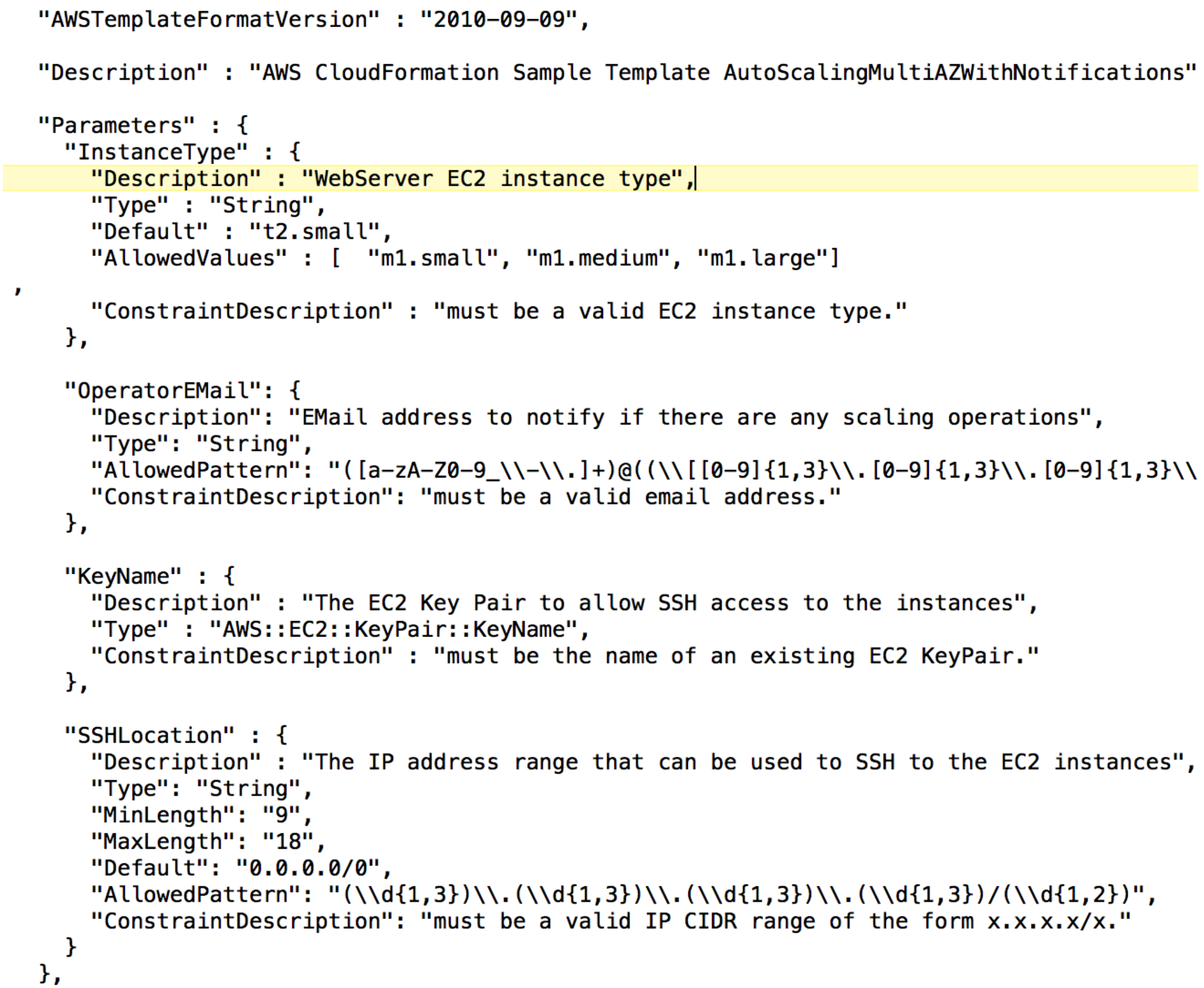
Speed

Ease

Security

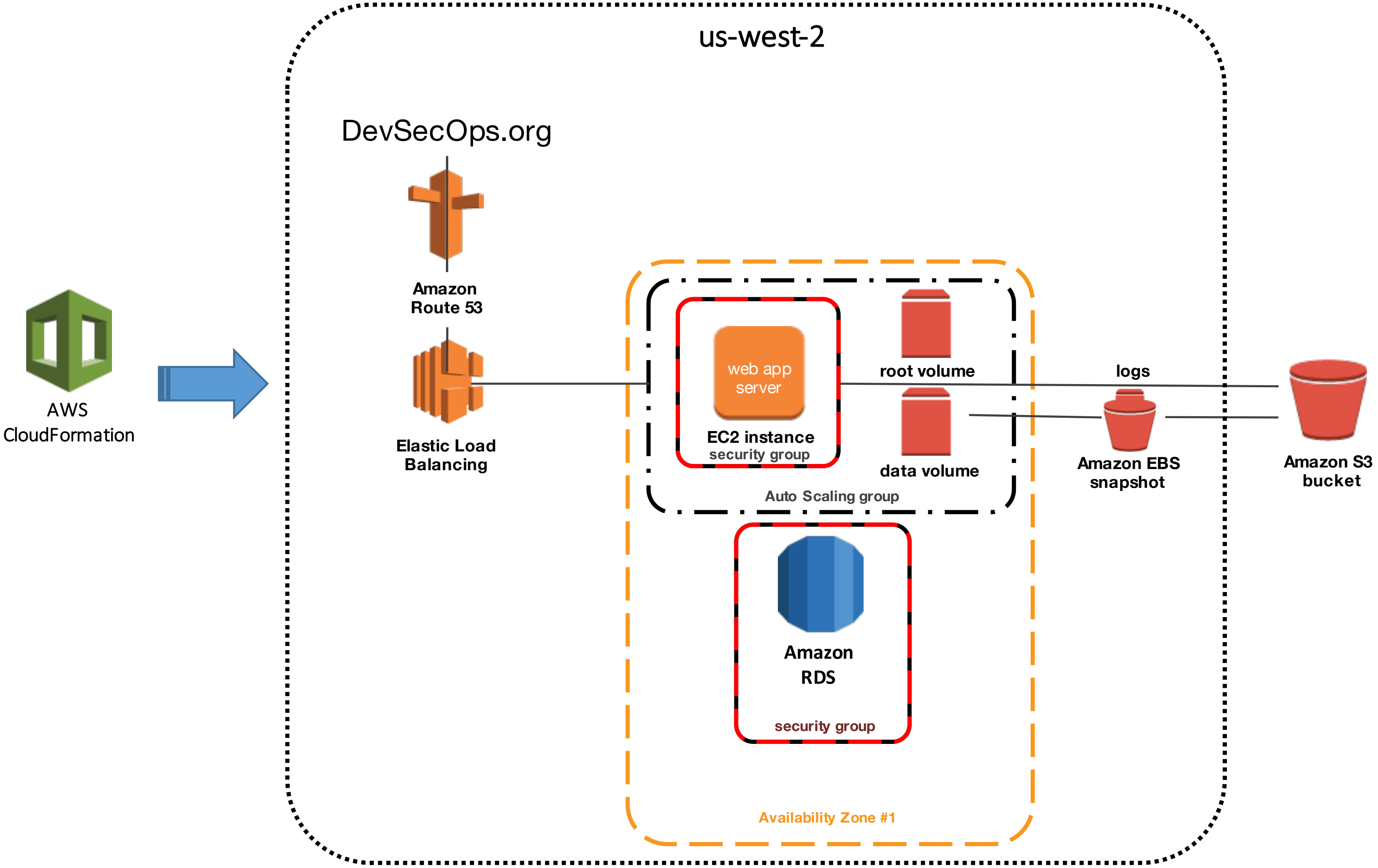
Skills Needed for SDE Development

- Networking
- Programming
- Application
- Security
- Operating System
- Micro-services



```
  "AWSTemplateFormatVersion" : "2010-09-09",

  "Description" : "AWS CloudFormation Sample Template AutoScalingMultiAZWithNotifications"

  "Parameters" : {
    "InstanceType" : {
      "Description" : "WebServer EC2 instance type",
      "Type" : "String",
      "Default" : "t2.small",
      "AllowedValues" : [  "m1.small", "m1.medium", "m1.large"]
,
      "ConstraintDescription" : "must be a valid EC2 instance type."
    },

    "OperatorEMail": {
      "Description": "EMail address to notify if there are any scaling operations",
      "Type": "String",
      "AllowedPattern": "([a-zA-Z0-9_\\-\\.]+)@((\\[[0-9]{1,3}\\.[0-9]{1,3}\\.[0-9]{1,3}\\
      "ConstraintDescription": "must be a valid email address."
    },

    "KeyName" : {
      "Description" : "The EC2 Key Pair to allow SSH access to the instances",
      "Type" : "AWS::EC2::KeyPair::KeyName",
      "ConstraintDescription" : "must be the name of an existing EC2 KeyPair."
    },

    "SSHLocation" : {
      "Description" : "The IP address range that can be used to SSH to the EC2 instances",
      "Type": "String",
      "MinLength": "9",
      "MaxLength": "18",
      "Default": "0.0.0.0/0",
      "AllowedPattern": "(\\d{1,3})\\.(\\d{1,3})\\.(\\d{1,3})\\.(\\d{1,3})/(\\d{1,2})",
      "ConstraintDescription": "must be a valid IP CIDR range of the form x.x.x.x/x."
    }
  },
```

SDE Example in AWS



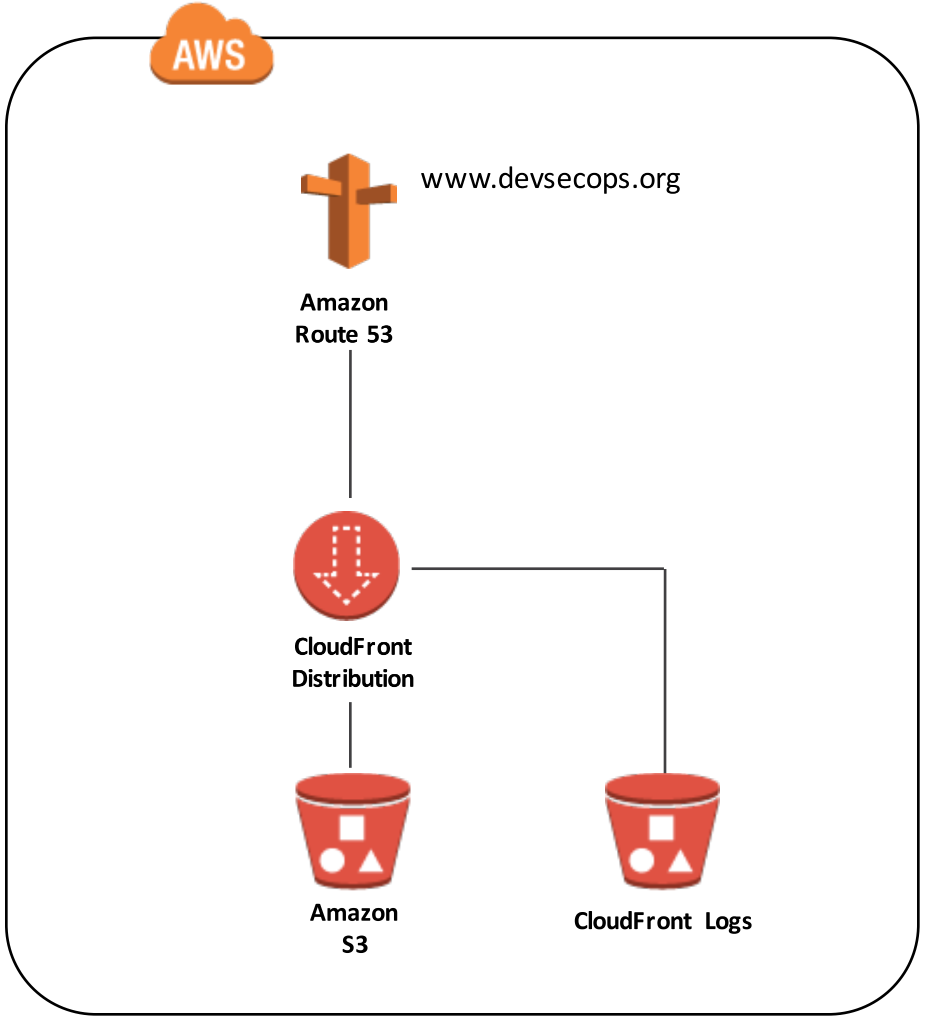
Micro-services

- Decomposed Applications where each piece of functionality is its own service
- Scales by replicating these microservices across computing resources as needed
- Usually use a light weight communication protocol (HTTPS API)
- Commonly leverages a queue

CC0 Public Domain

Serverless Microservice Architecture

- Managed services make it possible to run a full application that does not require physical server
- Each managed service is considered a microservice
- Multiple microservices can be put together to create a fully functional application
- Great for HTML 5 and Angular web applications



Serverless Microservice Architecture

- Amazon Web Services
- Google Cloud Platform
- Rackspace
- Heroku
- Many others

Security Considerations

- Is there visibility?
- Is there logging?
- Is there auditing the logging?
- Are there service logs?
- Are there API access logs?
- Is there encryption?
- Can customers control their own encryption keys?

Is everybody setup?

- Login in and familiarize yourselves with CentOS 7

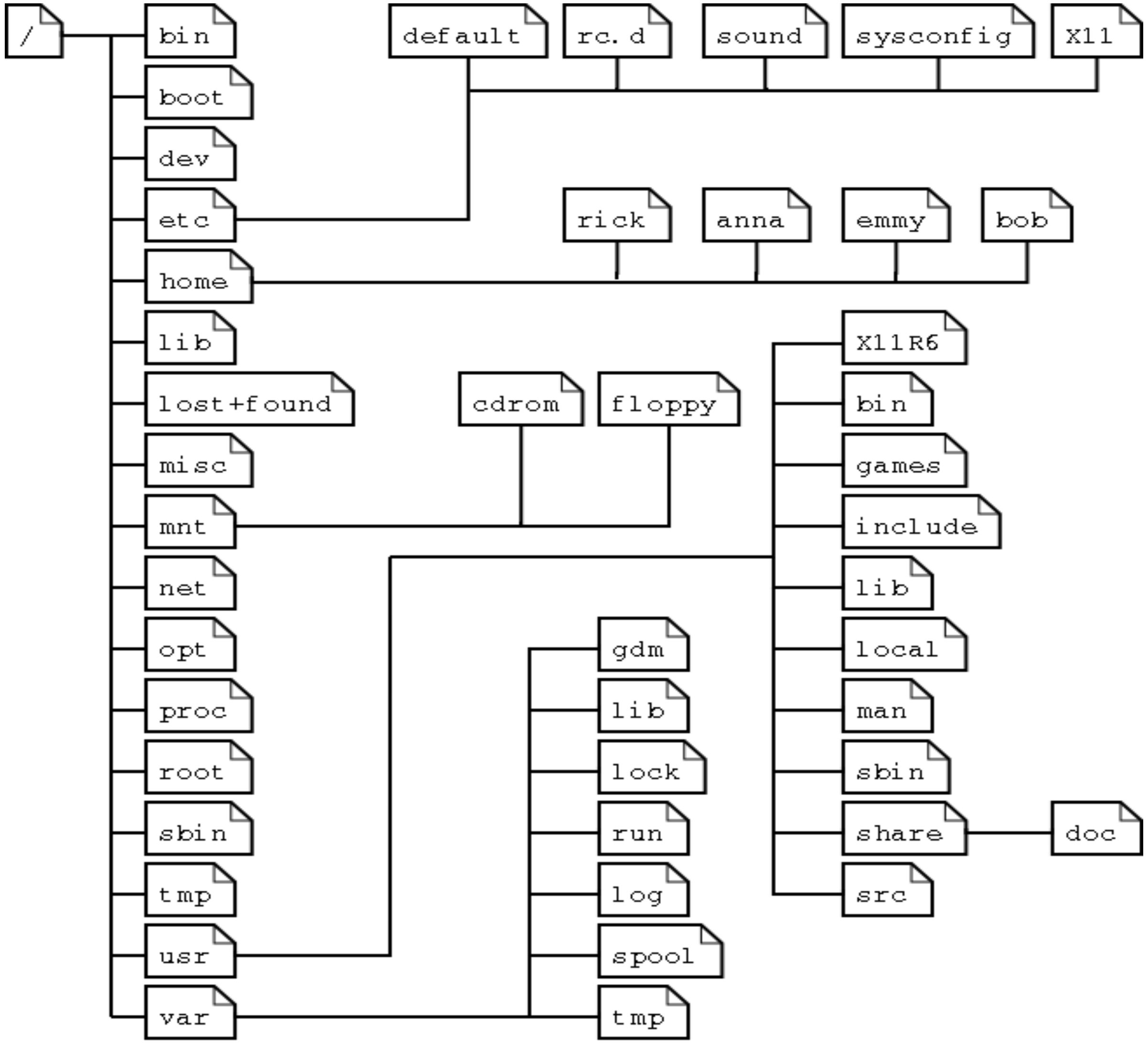
```
$ vagrant ssh
```

Lightning fast intro to Linux/CentOS 7

- Follow along...
  - sudo
  - yum
  - Commands: cd, ls, find, grep
  - The filesystem
  - The network

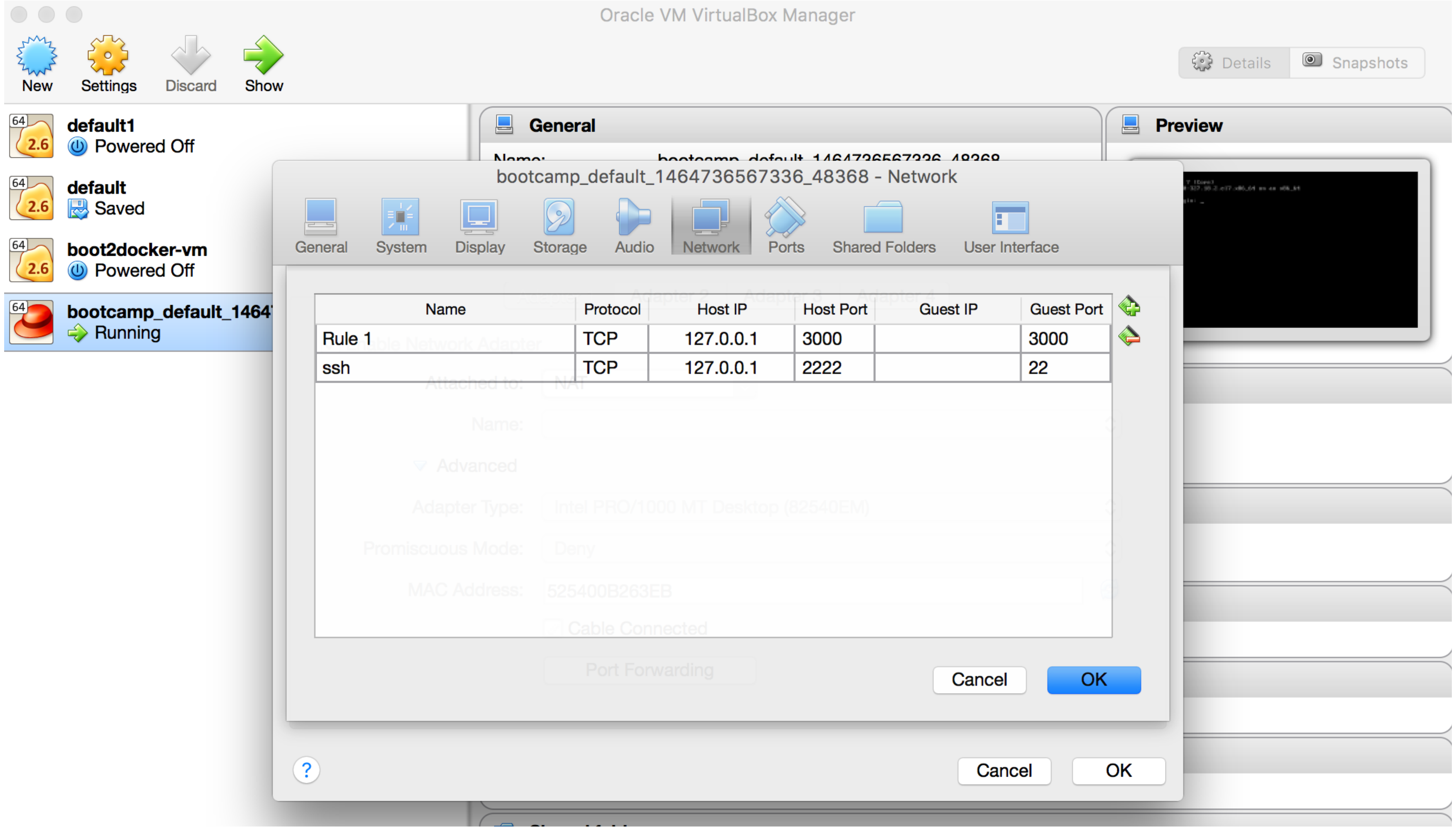
Lightning fast intro to Linux/CentOS 7

the filesystem:



http://www.tldp.org/

Port Forwarding (for the lab)



Name	Protocol	Host IP	Host Port	Guest IP	Guest Port
Rule 1	TCP	127.0.0.1	3000		3000
ssh	TCP	127.0.0.1	2222		22

Let’s get going

- Login in and familiarize yourself with rails and mariadb

```
$ vagrant ssh
$ sudo yum -y install mariadb mariadb-server
$ gem install rails
```

Lab 1 – Getting your Lab Environment Ready

- https://github.com/devsecops/bootcamp/blob/master/Week-2/labs/LESSON-1.md