

next  
session

sinisa  
sokolic



0	1	0	1	0
1	0	1	0	1
0	1	0	1	0
1	0	1	0	1
0	1	0	1	0
1	0	1	0	1
0	1	0	1	0
1	0	1	0	1
0	1	0	1	0
1	0	1	0	1
0	1	0	1	0
1	0	1	0	1
0	1	0	1	0

The CLOUD  
A Brave new World



# | about me



## sinisa sokolic

CTO @ RIS Consulting GmbH  
Citrix CTA

Cologne, Germany  
[www.sinisasokolic.com](http://www.sinisasokolic.com)

| talk  
about



**Enterprise IT**

**Startup IT**

**Do it Better**

# oldschool migrations



# traditional infrastructure “teams”

## Everybody knows these....

**Never touch a running system  
We have done it that way for years  
I'm too busy and have no time for changes  
I don't want to be responsible for that or I am not responsible  
Someone in the “team” will do the job  
Errors and failures are hidden whenever possible**

# about BOFH mindsets...

**Do you know why the system is slow?" they ask.  
"It's probably something to do with..." I look up today's excuse "... clock speed."  
"Oh" (Not knowing what I'm talking about, they're satisfied) "Do you know when it will be fixed?"  
"Fixed? There's 275 users on your machine, and one of them is you. Don't be so selfish - logout now and give someone else a chance!"**

## Waterfall Model

**Minimizing risk instead of maximizing agility**

**Restrict individual autonomy**

**Slows down feedback loops**

**Requires many teams and checkpoints for every small change**

**Lack of communication between the teams**

**Meeting marathons**

**Requirements gathered in a structured way and by the time a lot of them had changed**

Requirements

• 2014

Design

Implementation

• 2015

Verification

Maintenance

• 2017

We won't  
make it!

Crap

Fix it!!!

Somehow  
working

Throw more  
resources in...

...slows things  
more down.

It always takes longer than you expect,  
even when you take into account  
Hofstadter's Law.

— *Douglas Hofstadter, Gödel, Escher, Bach: An Eternal Golden Braid*<sup>[1]</sup>

# staging ...dreams



DEV



ACC



PRD

# Staging ...reality



DEV

ACC

PRD



## Old infrastructures...

- are complex
- hard to maintain
- have grown over decades
- are hard to change because of the dependencies they have
- are error prone
- do not use OpenSource software
- consist of too much silos
- change slowly

And the teams that run such an infrastructure seek for a reason to exist.



deep  
impact

# world map

AWS

Frankfurt, North America, Sao Paulo, Mumbai  
round about 600 servers

Netflix Stack

80% Linux, 20% Windows

Puppet, TeamCity, Bitbucket, Jira...  
about 250.000 EUR monthly costs



# A hard learning path...

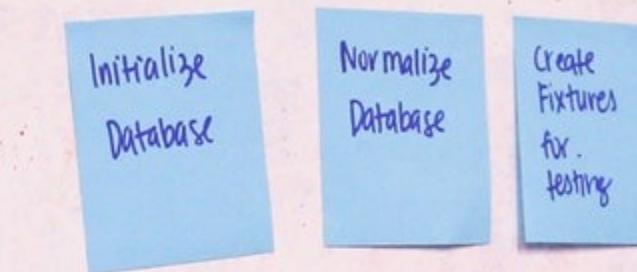
# WEEK 3 : Design



Infrastructure Recs:  
- Laravel + Backbone

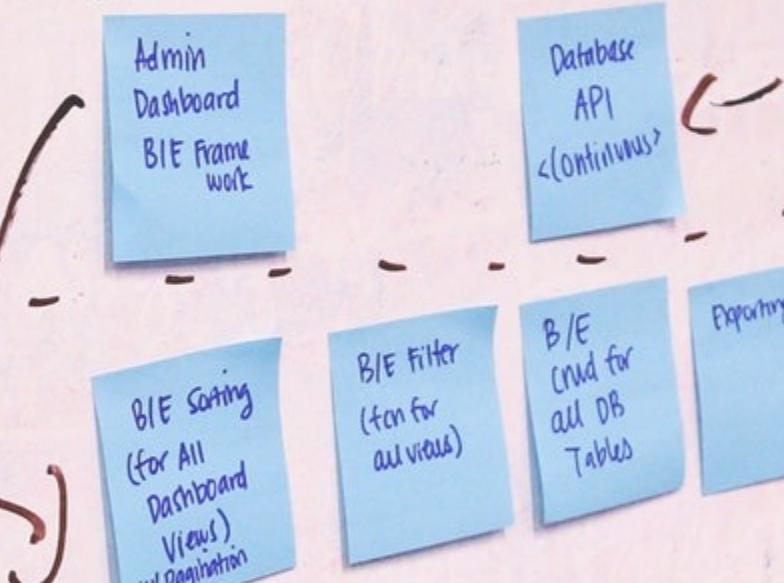
PLEASE DO NOT ERASE

# WEEK 4 : Design

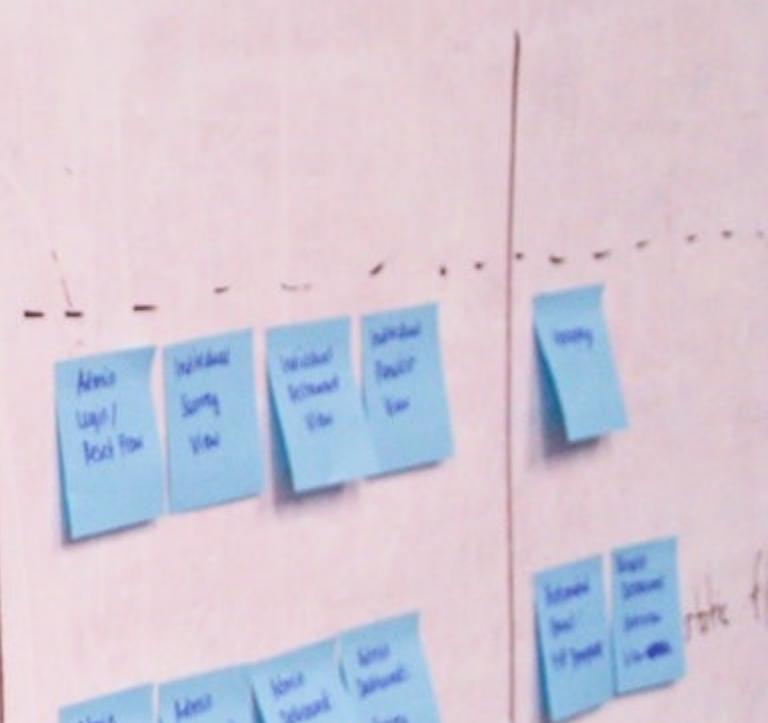


← Create Fixtures for testing

# WEEK 5 : Design



← Engineering



Developers and Scrum everywhere...



**Three weeks later I understood the basic infrastructure and how it was run...  
...developer driven.**



It felt like this...



**Phil Calçado** @pcalcado

Working for infrastructure startups in 2017

Expectation: "Come work with service mesh and Kubernetes and the bleeding edge development in infrastructure software"

Reality: "\$ man iptables"

do better

# conclusion 1

**Why startups suck from an enterprise perspective...**

Unstructured  
Bunch  
Of  
Developers  
And  
Freaks  
Doing  
Uncoordinated  
Stuff

## conclusion 2

# Why Enterprise IT sucks from a startup perspective...

Lame  
Old  
Dinosaurs  
Not  
Capable  
Of  
Delivering  
Infrastructure  
In  
Time

the  
solution?

**Better together...**

**As Startups grow they must transform to enterprise methodologies (some of them)**

**Enterprise Architects can help in the design of security, scalability and reproducibility**

**Learn agility from developers**

choose  
wisely...

**Assess your existing workloads before you move them 1:1 in the Cloud**

**Check existing available services (AWS / Azure)**

**Migrating the infrastructure 1:1 to the cloud might behave shitty...**

**...and might drive additional unnecessary costs**



Services

Resource Groups



EC2 Container Service



Microsoft Azure

## EC2 Dashboard

Events

Tags

Reports

Limits

## INSTANCES

Instances

Spot Requests

Reserved Instances

Dedicated Hosts

## IMAGES

AMIs

## Resources

You are using the following Amazon EC2 resources in the EU Central (Frankfurt) region:

52 Running Instances

0 Dedicated Hosts

84 Volumes

8 Key Pairs

0 Placement Groups

Just need a simple virtual private server? Get everything you need to jumpstart your project - compute, storage  
for free.

Create Instance



New

Dashboard

Resource groups

All resources

Recent

App Services

Virtual machines (classic)

Virtual machines

## Dashboard

Resources

RG-MAIN

Automation

Automation Account

West Europe

OMS

Recovery Services vault

West Europe

AzureActivity(OMS-RISC-Workspace)

Solution

West Europe

Gitlab-ip

Public IP address

West Europe

Gitlab

Virtual machine

West Europe

gitlab129

Network interface

West Europe

If you are using one of these consoles to operate your environment you are not using the full capabilities of the cloud.



# Infrastructure Operations

=

# Software Development

# Infrastructure as Code

## Benefits

- **Source controlled environment**
- **Change visibility**
- **Easy rollback to previous version**
- **Scale out**
- **Easily create new environments**
- **Blue/green deployments**
- **Replication**
  - **DEV/ACC/PRD**
  - **Azure/AWS/Google**
- **Self-healing**
- **Idempotency**

```
2466 "InstallServer":{  
2467     "Type": "AWS::EC2::Instance",  
2468     "DependsOn": "DomainControllerWaitCondition",  
2469     "Metadata":{  
2470         "AWS::CloudFormation::Init":{  
2471             "configSets":{  
2472                 "config": [  
2473                     "setup",  
2474                     "rename",  
2475                     "join",  
2476                     "share",  
2477                     "install"  
2478                 ]  
2479             },  
2480             "setup":{  
2481                 "sources":{  
2482                     "C:\\\\users\\\\public\\\\downloads\\\\": "https://s3.amazonaws.com/cf-XenApp/ag_ctxcloud_com.zip"  
2483                 },  
2484                 "files":{  
2485                     "c:\\\\cfn\\\\cfn-credentials":{  
2486                         "content":{  
2487                             "Fn::Join": [  
2488                                 "",  
2489                                 [  
2490                                     "AWSAccessKeyId=",  
2491                                     {  
2492                                         "Ref": "IAMUserAccessKey"  
2493                                     },  
2494                                     "\n",  
2495                                     "AWSSecretKey=",  
2496                                     {  
2497                                         "Fn::GetAtt": [  
2498                                             "IAMUserAccessKey",  
2499                                             "SecretAccessKey"  
2500                                         ]  
2501                                     },  
2502                                     "\n"  
2503                                 ]  
2504                             }  
2505                         },  
2506                     "c:\\\\cfn\\\\cfn-hup.conf":{  
2507                         "content":{  
2508                             "Fn::Join": [  
2509                                 "",  
2510                                 [  
2511                                     "[main]\\n",  
2512                                     "stack=",  
2513                                     {  
2514                                         "Ref": "AWS::StackName"  
2515                                     },  
2516                                     "\n",  
2517                                     "credential-file=c:\\\\cfn\\\\cfn-credentials\\n",  
2518                                     "region=",  
2519                                     {  
2520                                         "Fn::Sub": [  
2521                                             "arn:aws:iam::${AWS::AccountId}:access-key",  
2522                                             "Fn::GetAtt": [  
2523                                                 "AWS::StackName",  
2524                                                 "Arn"  
2525                                             ]  
2526                                         ]  
2527                                     }  
2528                                 ]  
2529                             }  
2530                         }  
2531                     }  
2532                 }  
2533             }  
2534         }  
2535     }  
2536 }
```



31,341 Running 63 Branches Tags

	Pipeline	Commit	Stage
running	#5233451 by 🚀 latest	↵ <b>20916-issues-mrs-linel...</b> -o e3612f86 🚀 Add content_class for limited width	✓
running	#5233393 by 🚀 latest	↵ <b>fix/refactor-cycle-ana...</b> -o ab98279e 🚀 big refactor based on MR feedback	✓
running	#5233299 by 🚀 latest	↵ <b>dz-nested-groups</b> -o 2b2b39f0 🚀 Add support for nested namespaces in th...	✓
running	#5233284 by 🚀 latest	↵ <b>master</b> -o 2f91c0ee 🚀 Merge branch 'cleanup-common_utils.js'...	✓
running	#5233164 by 🚀 latest	↵ <b>25106-hide-issue-mr-bu...</b> -o b3d15756 🚀 adds changelog	✓
canceled	#5233032 by 🚀	↵ <b>25106-hide-issue-mr-bu...</b> -o 2fedc86e 🚀 adds check for logged in user in group iss...	✗
running	#5232996 by 🚀 latest	↵ <b>25209-improve-length-v...</b> -o 62d5439e 🚀 Use :maximum instead of :within for leng...	✓
running	#5232985 by 🚀	↵ <b>master</b> -o af19d56f	✗

# Version everything

## Benefits of Source Code Management

- It's totally free
- Track each change by contributor
- Prevent conflicting work
- "see" the changes
- Comment changes
- Easy roll-back
- Create parallel work streams

<https://www.atlassian.com/git/tutorials/atlassian-git-cheatsheet>

## Continuous Integration

In **software engineering**, continuous integration (CI) is the practice of merging all developer working copies to a shared **mainline** several times a day.

## Continuous Delivery

Continuous delivery (CD) is a **software engineering** approach in which teams produce software in short cycles, ensuring that the software can be reliably released at any time

## Continuous Testing

Continuous testing is the process of executing **automated tests** as part of the software delivery pipeline to obtain immediate feedback on the business risks associated with a software release candidate

## Continuous Deployment

Continuous deployment means that every change is automatically deployed to production. Continuous delivery means that the team ensures every change can be deployed to production but may choose not to do it, usually due to business reasons.

# Tools and processes in DevOps...

**Individuals and interactions over processes and tools**

**But these make our life complicated... Way more complicated!**

**Align Dev Tools and Ops Tools to have a base to work on together**

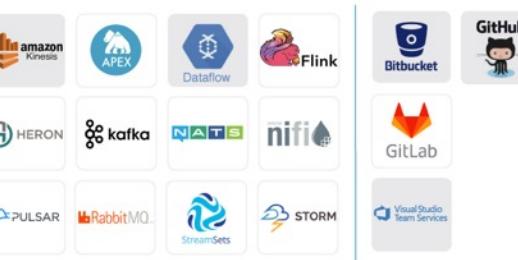
# Cloud Native Landscape

v0.9.9

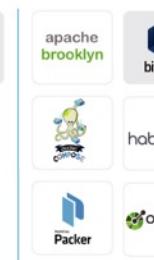
## Database & Data Analytics



## Streaming



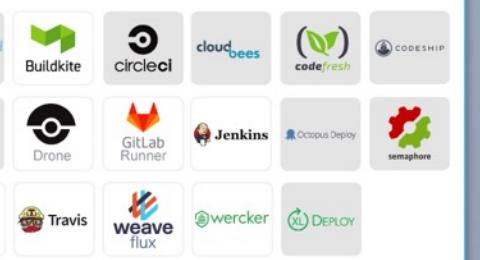
## SCM



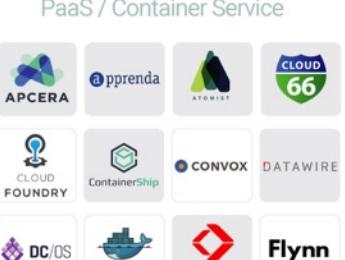
## Application Definition



## CI/CD



## Platforms

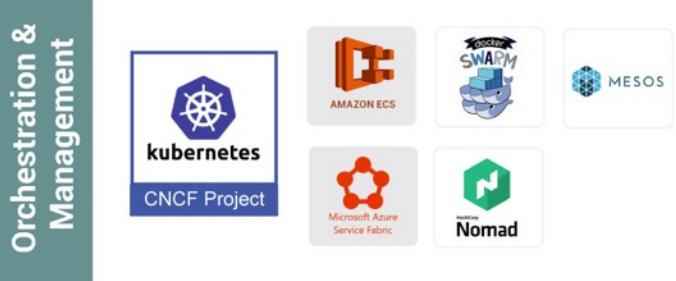


## Observability & Analysis

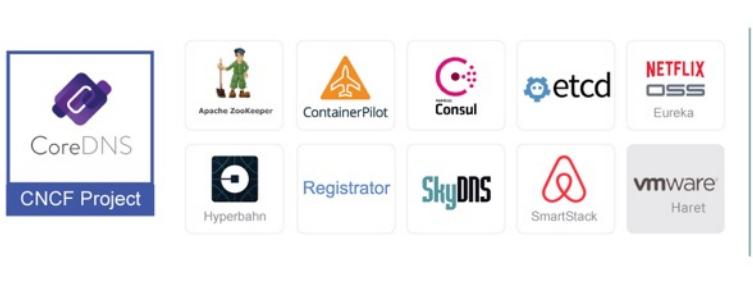
### Monitoring



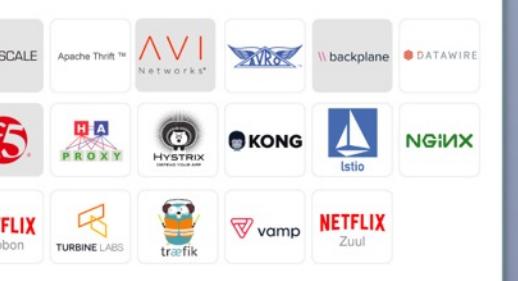
## Scheduling & Orchestration



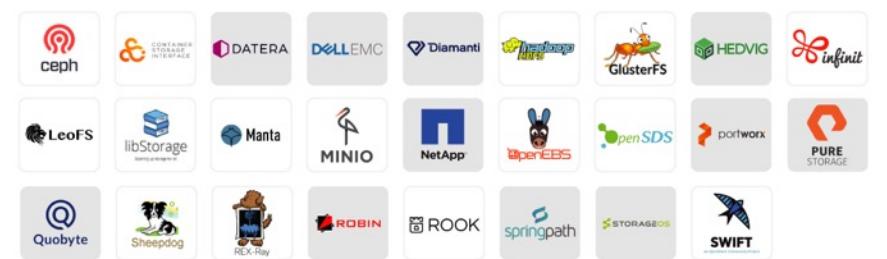
## Coordination & Service Discovery



## Service Management



## Cloud-Native Storage



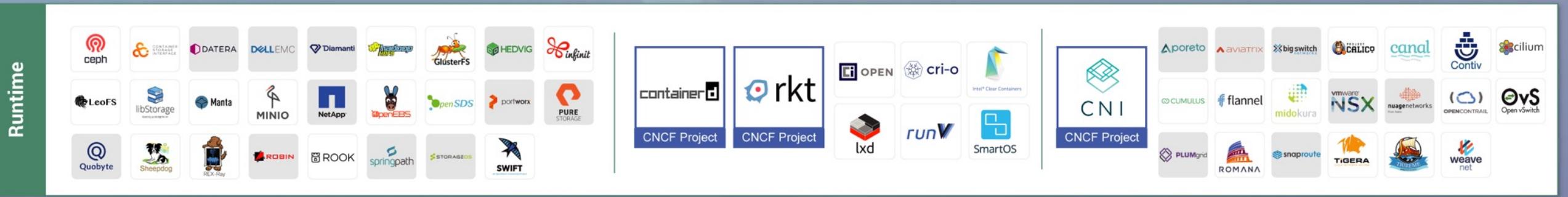
## Container Runtime



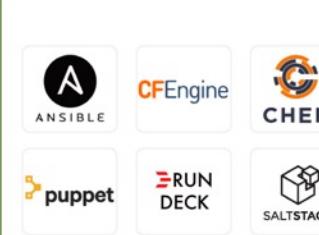
## Cloud-Native Network



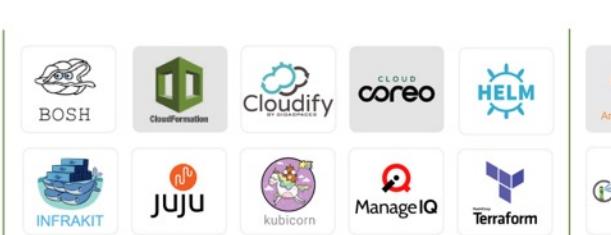
## Runtime



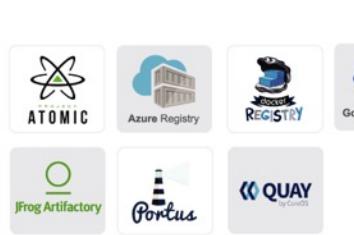
## Host Management / Tooling



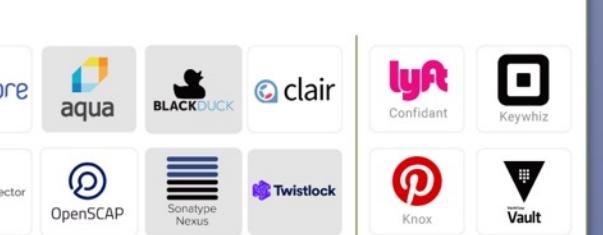
## Infrastructure Automation



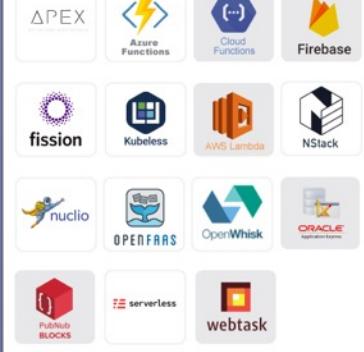
## Container Registries



## Secure Images



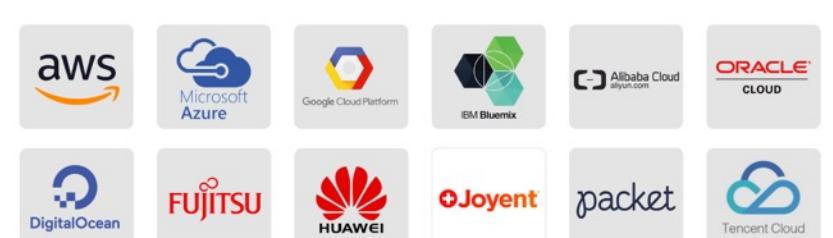
## Key Management



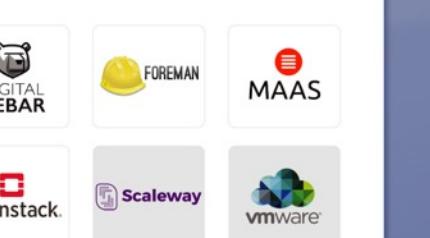
## Provisioning



## Public



## Private



## Cloud



[github.com/cncf/landscape](http://github.com/cncf/landscape)



Greyed logos are not open source

This landscape is intended as a map through the previously uncharted terrain of cloud native technologies. There are many routes to deploying a cloud native application, with CNCF Projects representing a particularly well-traveled path.

**CLOUD NATIVE COMPUTING FOUNDATION**

**Redpoint** | **Amplify**

# Good coders code, great coders reuse.

<https://github.com/hashicorp/terraform.git>

<https://github.com/hashicorp/packer.git>

<https://github.com/puppetlabs/puppet.git>

<https://github.com/puppetlabs/r10k.git>

FILTER BY

Provider ▾

  Verified

**lb-http**  
google   
Modular Global HTTP Load Balancer for GCE using forwarding rules.  
  
Version 1.0.4 · By GoogleCloudPlatform

**vpc**  
aws   
Terraform module which creates VPC resources on AWS  
  
Version 1.0.4 · By terraform-aws-modules

**managed-instance-group**  
google   
Modular Google Compute Engine managed instance group for...  
  
Version 1.1.3 · By GoogleCloudPlatform

**lb-internal**  
google   
Modular Internal Load Balancer for GCE using forwarding rules.  
  
Version 1.0.4 · By GoogleCloudPlatform

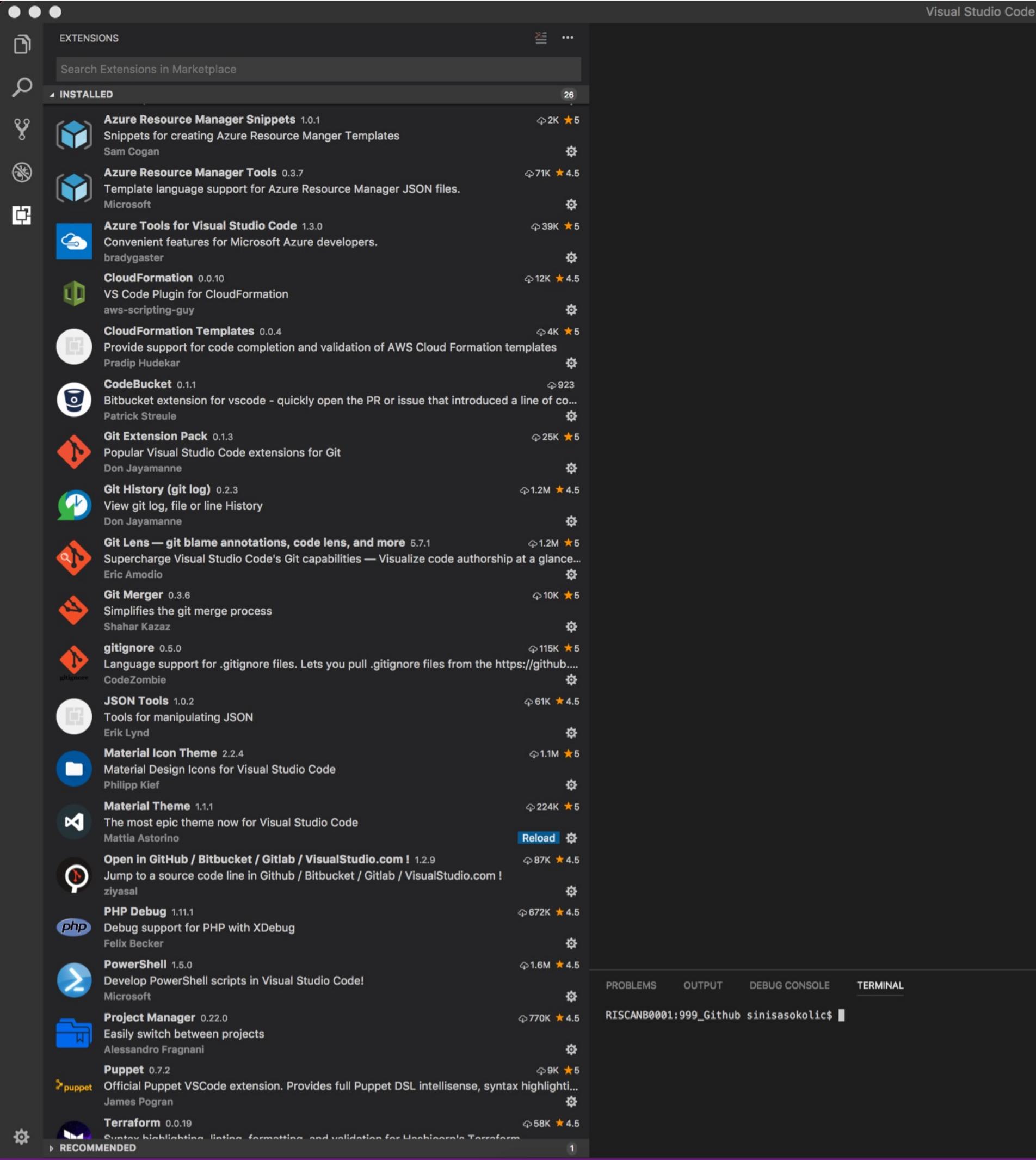
**nat-gateway**  
google   
Modular NAT Gateway on Google Compute Engine for Terraform.  
  
Version 1.0.5 · By GoogleCloudPlatform

**loadbalancer**  
azurerm   
Terraform Azure RM Module for Load Balancer  
  
Version 1.0.1 · By Azure

**network**  
azurerm   
Terraform Azure RM Module for Network  
  
Version 1.1.0 · By Azure

**computegroup**  
azurerm   
Terraform Azure RM Compute Group Module  
  
Version 1.0.1 · By Azure

**compute**  
azurerm   
Terraform Azure RM Compute Module  
  
Version 1.1.0 · By Azure



# The right console...

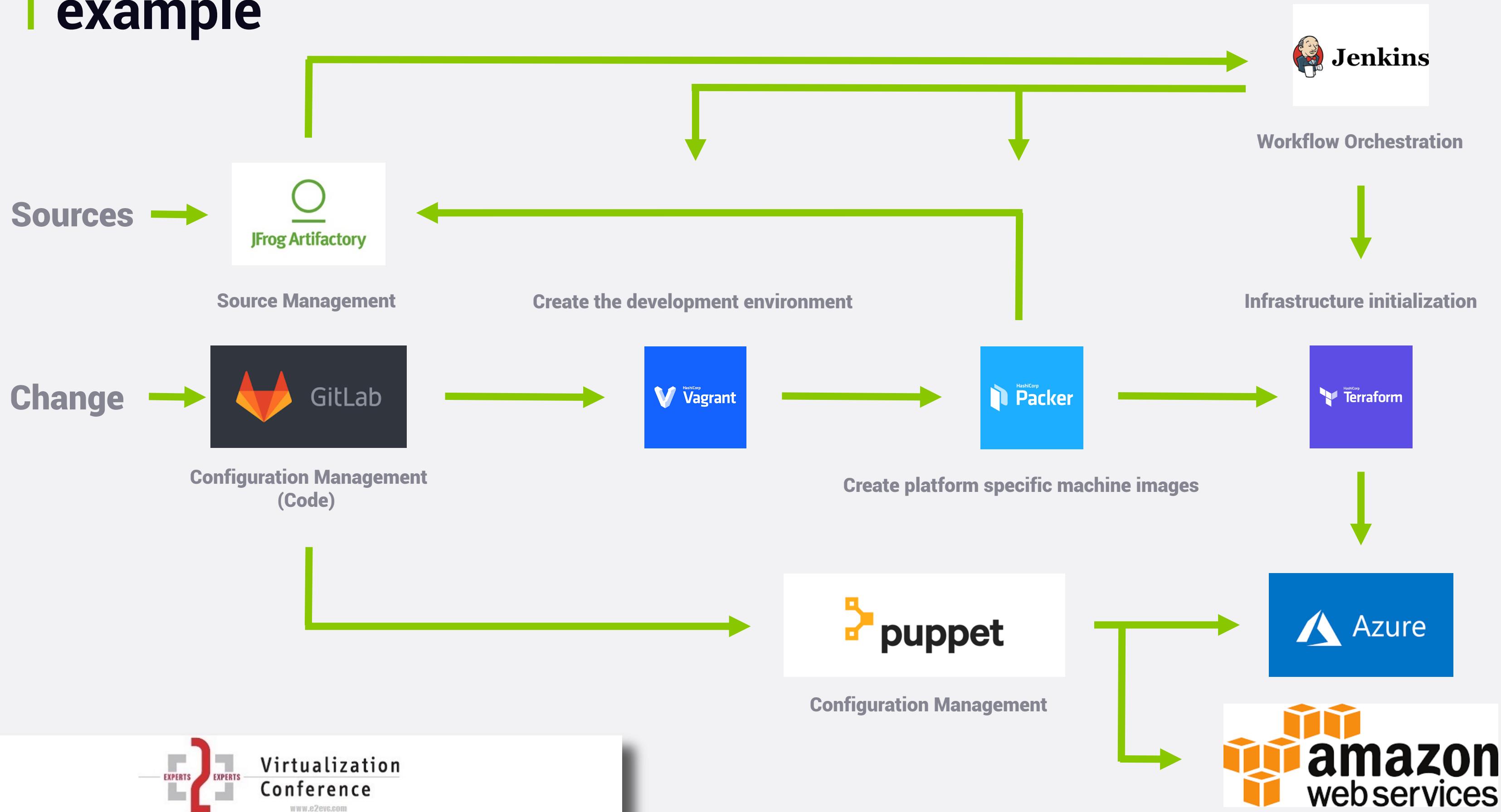
Supports most of the tools

- Git
- Terraform
- Cloudformation
- Azure
- PowerShell
- JSON
- YAML
- Puppet
- Terminal PS or Bash
- ...

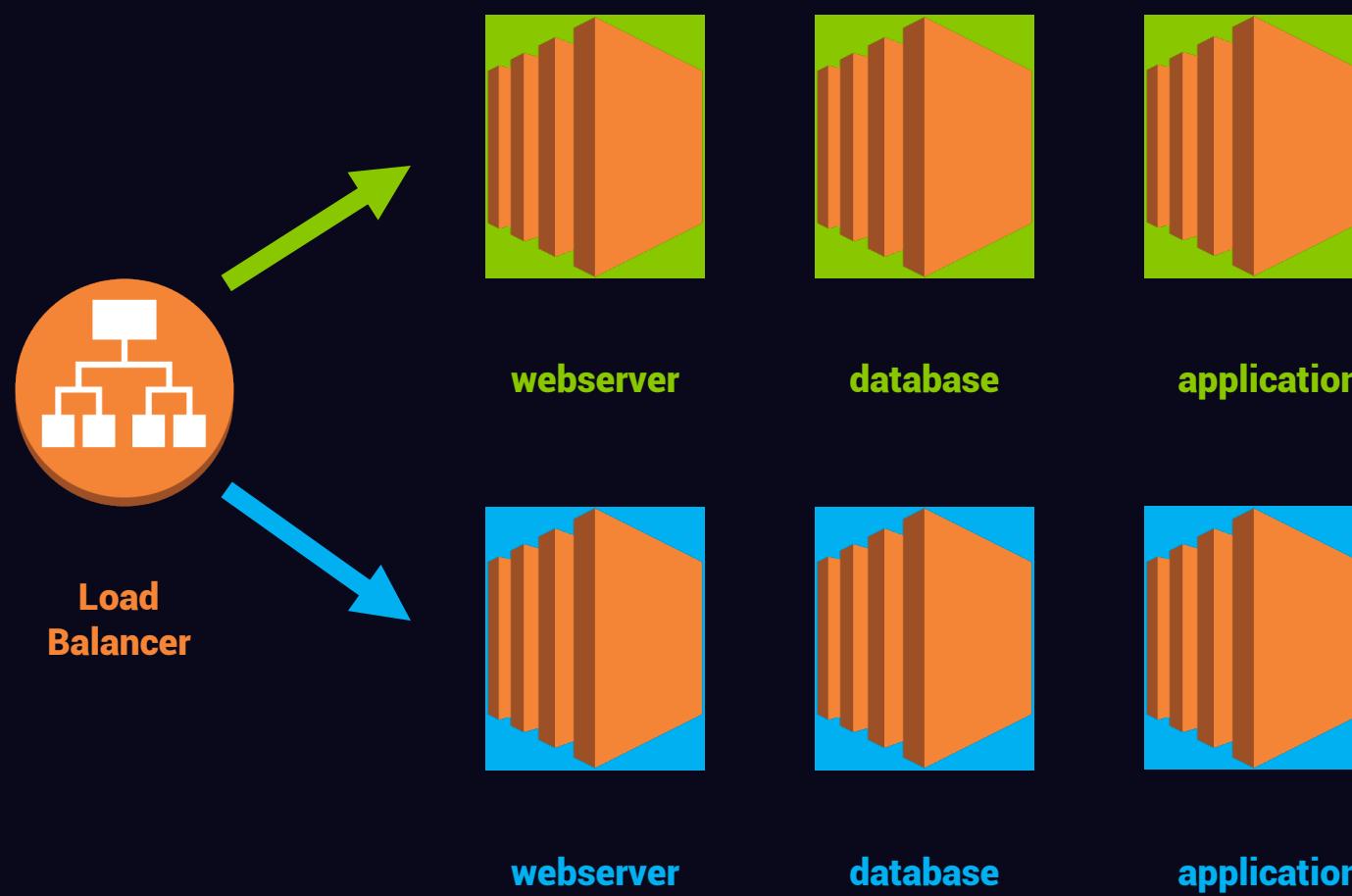
<https://code.visualstudio.com>

page  
034

# workflow example



# trash your deployments...

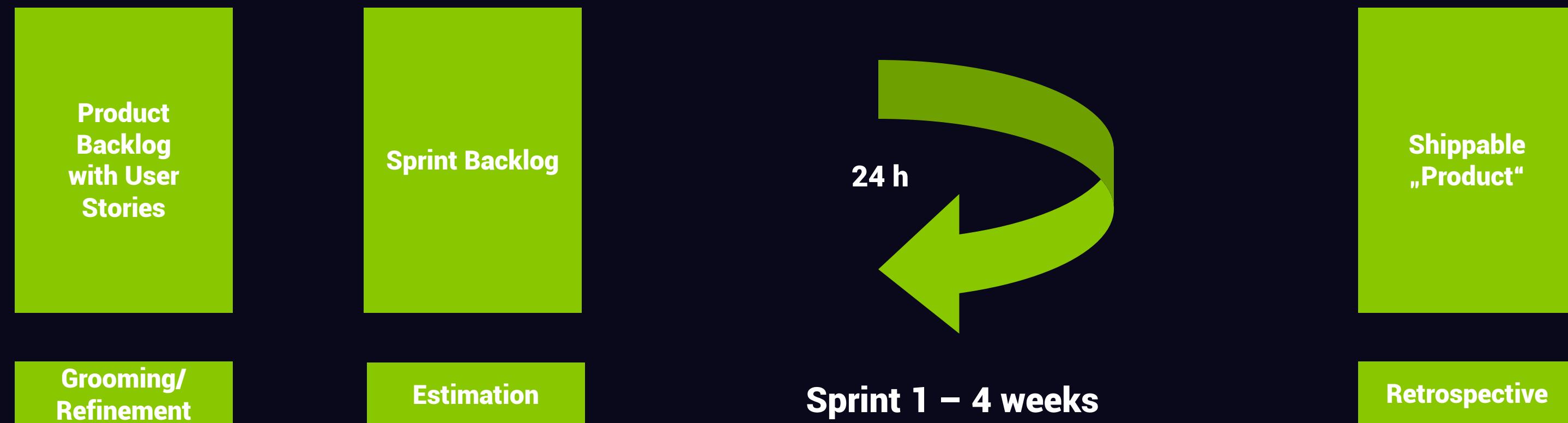


## Blue / Green Deployments

- Leverage cloud capabilities
- Continuous Integration and Deployment without down-times
- Fast roll-back
- Runs perfectly with IaC
- On-Demand hot-standby
- Nobody wants to work on weekends

# Scrumificate your team

Scrum is a framework for managing software development. It is designed for teams of three to nine developers who break their work into actions that can be completed within fixed duration cycles (called "sprints"), track progress and re-plan in daily 15-minute stand-up meetings, and collaborate to deliver workable software every sprint.<sup>[1][2]</sup> Approaches to coordinating the work of multiple scrum teams in larger organizations include Large-Scale Scrum, Scaled Agile Framework (SAFe) and Scrum of Scrums, among others. - Wikipedia



# Think about

- Docker
- Microservices
- Automatic Testing (Unit tests)
- DevOps, DevSecOps, DevBizOps...
- It's not for everybody
- What are your constraints or bottlenecks?
- Innovation
- Start to change

You can't directly change culture. But you can change behavior and behavior becomes culture.

- Lloyd Taylor – VP Infrastructure, Ngmoco

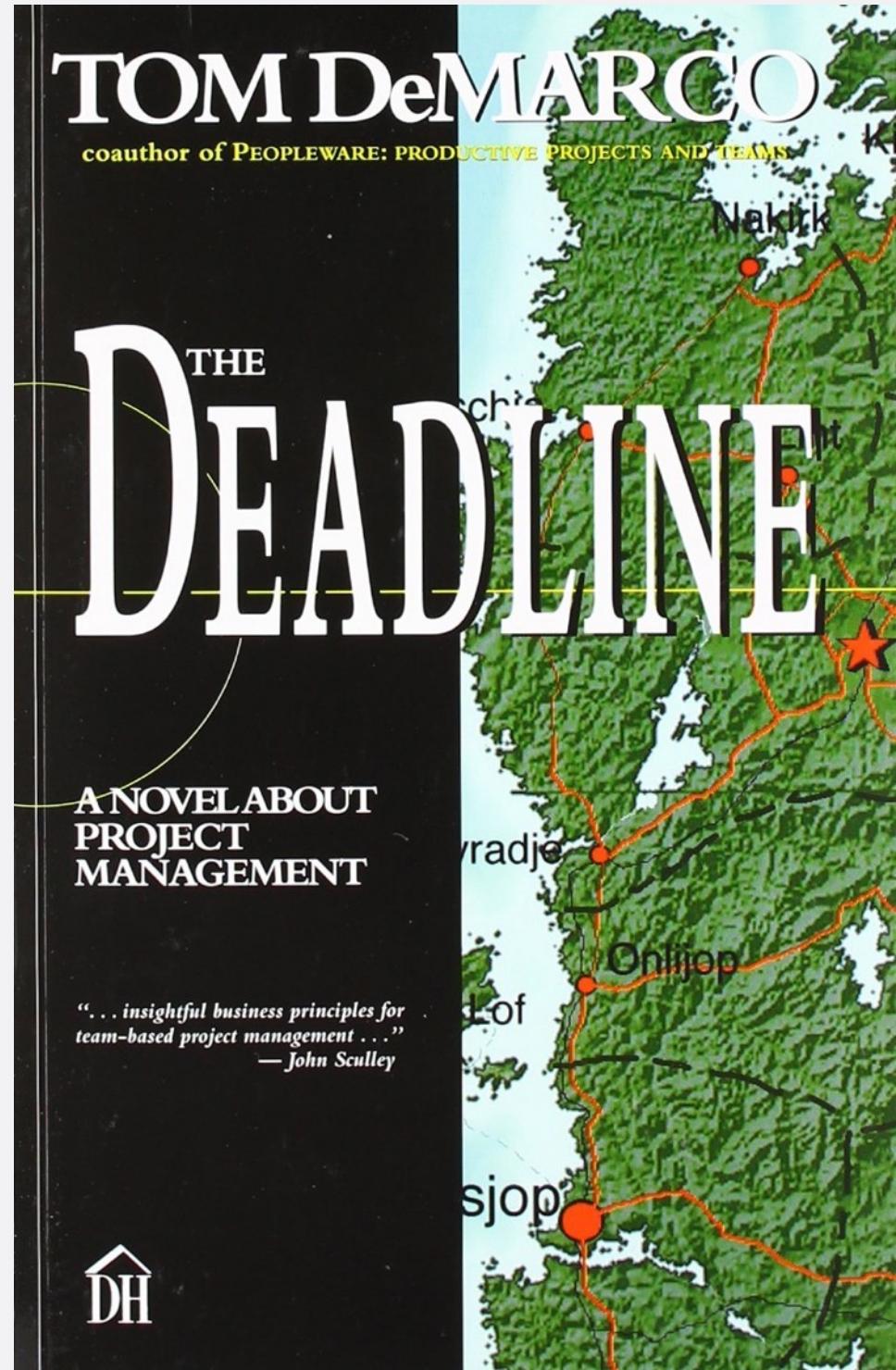
only the **BRAVE**  
survive



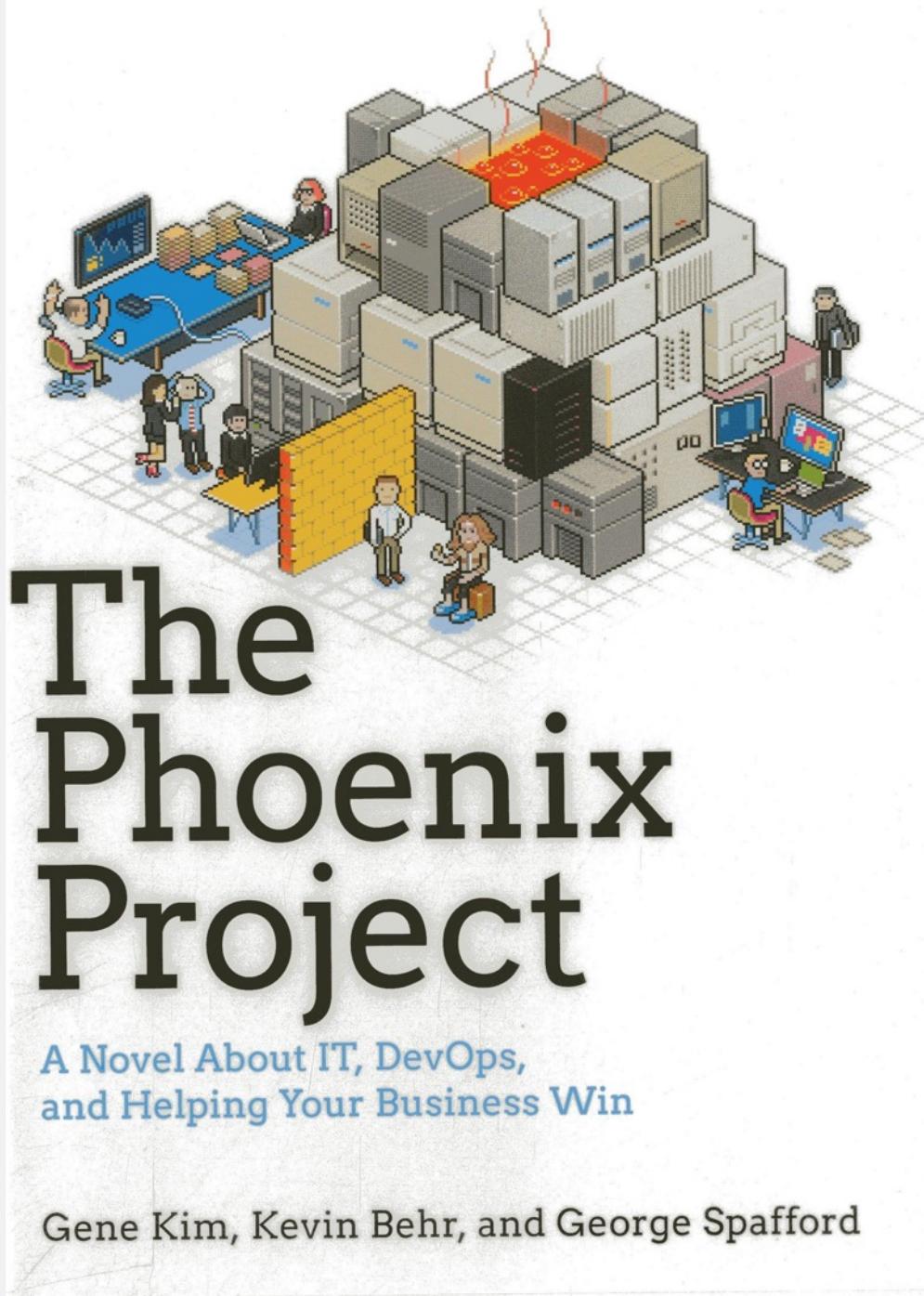
<https://github.com/Netflix/chaosmonkey>

**Chaos Monkey randomly terminates virtual machine instances and containers that run inside of your production environment. Exposing engineers to failures more frequently incentivizes them to build resilient services.**

# good reads



From the authors of *The Visible Ops Handbook*



thank you.

@sinisasokolic