Desmistificando Microsserviços e DevOps: Projetando Arquiteturas Efetivamente Escaláveis

Prof. Vinicius Cardoso Garcia
vcg@cin.ufpe.br :: @vinicius3w :: assertlab.com

[IF1004] - Seminários em SI 3
https://github.com/vinicius3w/if1004-DevOps

Licença do material

Este Trabalho foi licenciado com uma Licença

Creative Commons - Atribuição-NãoComercial-Compartilhalgual 3.0 Não Adaptada

Mais informações visite

http://creativecommons.org/licenses/by-nc-sa/3.0/deed.pt

Moving into the Future

Warm up

- How DevOps might evolve over the next few years?
- First, we defined DevOps as a collection of processes specific to the goal of reducing the time between a commit and the code being in normal production
- DevOps has been evolving quickly: organizational structure, process definition, and technology

Operations as a Process

If you can’t describe what you are doing as a process, you don’t know what you’re doing.
— W. Edwards Deming

Introduction

- The continuous deployment pipeline is not just another software product with system-of-systems characteristics, it also has strong characteristics of a process
- What are the implications of treating operations as processes?

Introduction

- You can discover a process model from existing operations software/scripts and their logs
  - Analyze the discovered process models for improvement opportunities
  - Use the process models to monitor the progression of various operations, to detect errors and to recover from them as early as possible
  - Set monitoring thresholds at a level that reflects the active operations processes. You want to achieve this at the step level rather than the whole process level, since that leads to earlier error detection and recovery
  - Use the process models to help other activities such as root cause diagnosis

Motivation and Overview

- Reliability refers to the capability of the overall deployment pipeline and its individual pieces to maintain service provision for defined periods of time (more informations here)
  - A typical pipeline has to deal with different types of error responses from different types of systems

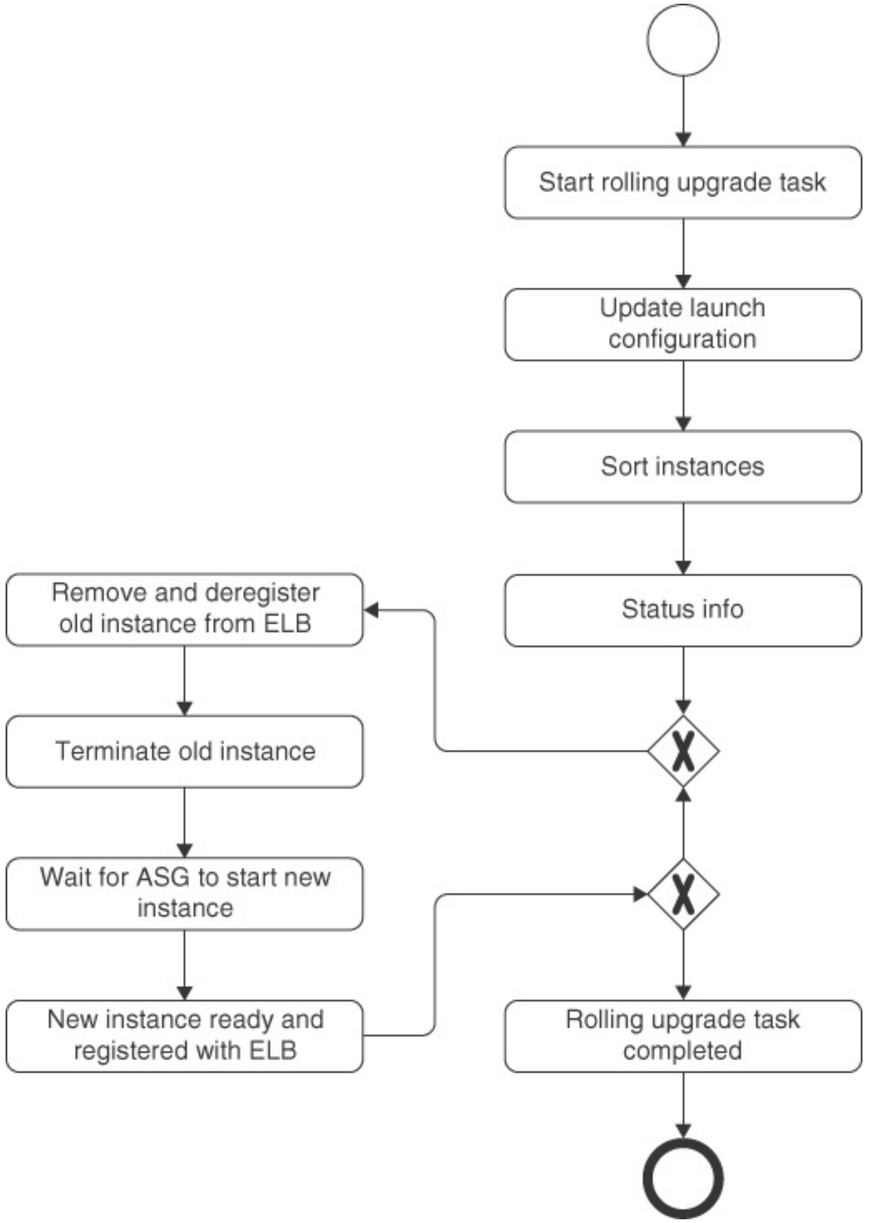
Features of Operation Processes

- Operations processes manipulate only a few types of entities
- Operations processes have a time frame measured in tens of minutes if not in hours
- Operations tools typically generate high-quality logs that can be used to create the process model without a lot of noise

Process Model

- We discover the process model by analyzing logs from successful executions of the process.
  - This discovery is done offline after we have achieved successful execution and generated associated logs
- We compare the desired state of the process with the current state of the process. Any difference indicates a reliability problem and provides the seeds of a recovery strategy
  - These activities happen online (concurrently) with the process

Example: process of rolling upgrade



Offline Activities

- The process model is created offline
- It can be created manually based on your understanding of the operation and the code/scripts
  - Alternatively, process mining techniques can be used to discover the process, especially from logs

Example: process of rolling upgrade

```
"2014-05-26_13:17:36 Started on thread Task:Pushing
ami-4583197f into group testworkload-r01 for app
testworkload."
"2014-05-26_13:17:38 The group testworkload-r01 has 8
instances. 8 will be replaced, 2 at a time."
"2014-05-26_13:17:38 Remove instances [i-226fa51c] from
Load Balancer ELB-01"
"2014-05-26_13:17:39 Deregistered instances [i-226fa51c]
from load balancer ELB-01"
"2014-05-26_13:17:42 Terminating instance i-226fa51c"
...
"2014-05-26_13:17:43 Waiting up to 1h 10m for new
instance of testworkload-r01 to become Pending."
```

Process Model discovery

- Process mining techniques allow the discovery of a process model from these logs without having access to the source code
  1. group the logs based on the activity they represent and tag them with an activity name, and
  2. use the tagged logs to create the process model using a tool such as ProM

Example: Sample CloudTraillog

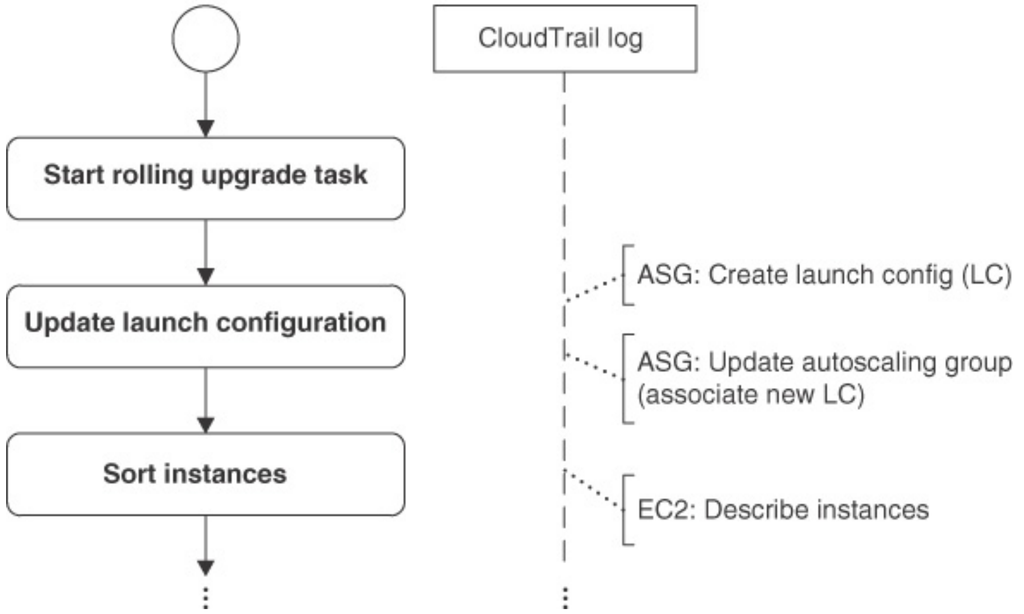

```
{ "awsRegion": "us-west-2",
  "eventName": "TerminateInstances",
  "eventSource": "ec2.amazonaws.com<http://ec2.amazonaws.
com>",
  "eventTime": "2014-01-24T01:59:58Z",
  "eventVersion": "1.0",
  "requestParameters": {"instancesSet": {"items":
[{"instanceId": "i-5424a45c"}]}},
    "responseElements": {"instancesSet": {"items":
[{"currentState": {"code": 32,"name": "shutting-
down"},"instanceId": "i-5424a45c","previousState":
{"code": 32,"name": "shutting-down"}}]}},

"sourceIPAddress":"autoscaling.amazonaws.com<http://
autoscaling.amazonaws.com>",
    "userAgent": "autoscaling.amazonaws.com<http://
autoscaling.amazonaws.com>",
    "userIdentity": {"accountId": "066611989206","arn":
"arn:aws:iam::066611989206:root","invokedBy":
"autoscaling.amazonaws.com<http://autoscaling.amazonaws.
com>","principalId": "066611989206","type": "Root"}}
```

Example: process of rolling upgrade

- Correlating CloudTrail logs with the process model to determine the AWS resources manipulated by each activity of the process model

- Knowing the state of these resources at the beginning of an activity and knowing the type of manipulations that should occur allows you to determine the expected state of the AWS resources at the end of each activity



Human interaction

- There are two steps in the development of the process model that require human intervention.
  - A human must examine the groups of activities to determine whether they are at a desired level of granularity and to assign names to the groups
  - A human must also examine the generated process model. It is possible that there are spurious activities or transitions within the process model. A human must determine that the model, in fact, represents the process being modeled

Online Activities

- Error detection and recovery can be done online during the execution of the rolling upgrade
- Diagnosis is an activity that occurs subsequent to the detection of an error, which can be online or offline;
  - Fast online diagnosis can lead to more informed recovery, but detailed analysis of underlying issues are better done offline, after recovery brought the system back to a stable state

Error Detection

- From the log lines being produced, we can detect the start and end of each activity step.
- From the process model, we know the desired sequence of steps
- One error detection mode is to look for steps out of the desired sequence. Such an occurrence is called a “conformance error”

Types of erros

- Unknown: a log line that is completely unknown
- Error: a log line that corresponds to a known error
- Unfit: a log line that corresponds to a known activity, but that should not happen given the current execution state of the process instance. This can be due to skipped activities (going forward in the process) or undone activities (going backward)

Example

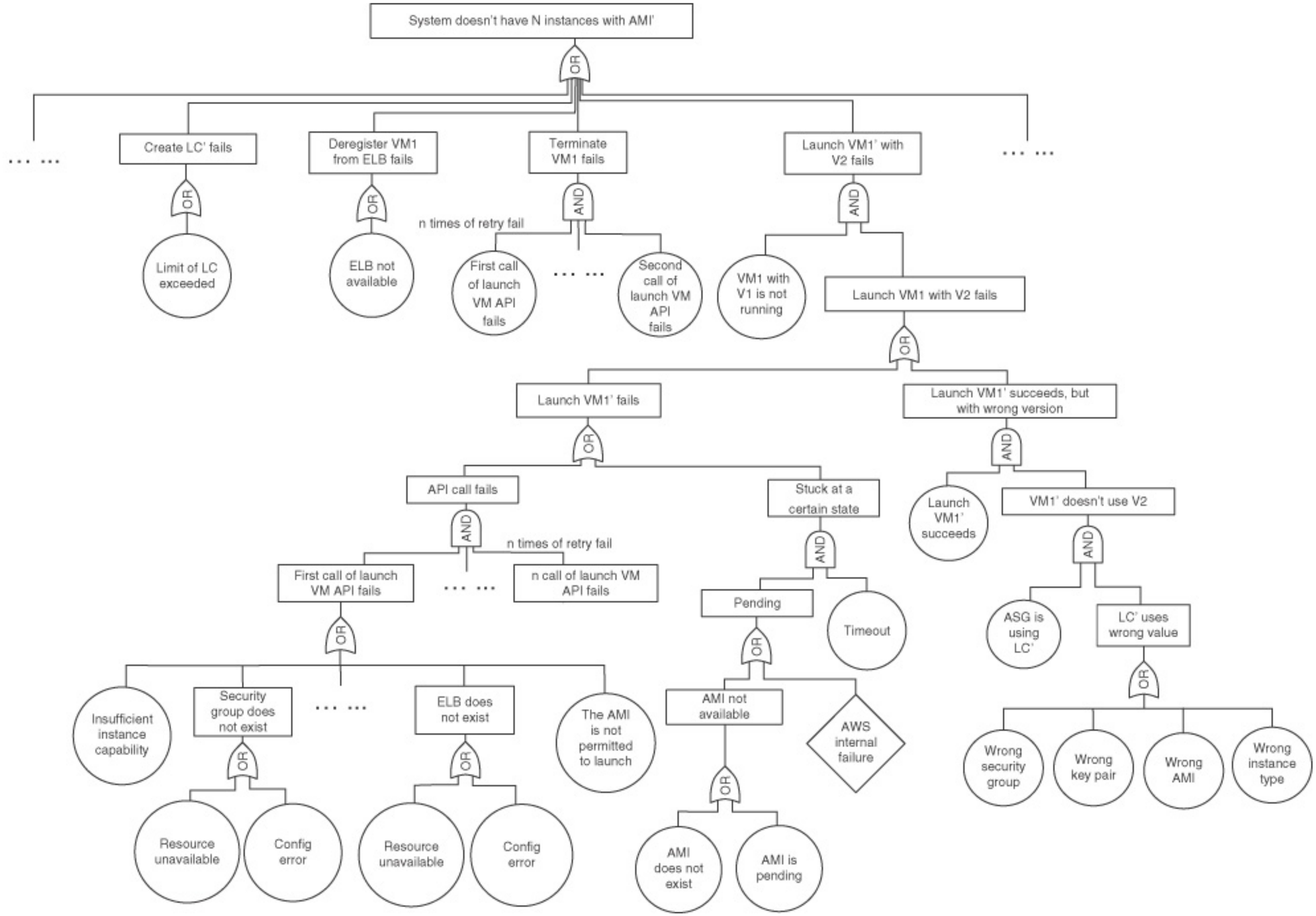
- "2014-05-26_13:17:38 Remove instances [i-226fa51c] from Load Balancer ELB-01", we should expect a log line about terminating that instance [i-226fa51c] soon according to the discovered process model

Error Recovery

- We have three sets of states of the AWS resources that are relevant
  - The state at the beginning of the last activity
  - The current erroneous state
  - The desired state
- We know that the current state is erroneous for some reason. There are at least two options to automatically recover from the error:
  - roll back to the state at the beginning of the last activity or roll forward to the desired state

Error Diagnosis

- Repairing an error may not repair the root cause of an error
- Diagnosing bugs in software is certainly important and useful
- For diagnosing operations errors, we use fault trees as a reference model


System doesn't have N instances with AMI'
OR
Create LC' fails
Deregister VM1 from ELB fails
Terminate VM1 fails
Launch VM1' with V2 fails
OR
AND
n times of retry fail
Limit of LC exceeded
ELB not available
First call of launch VM API fails
Second call of launch VM API fails
VM1 with V1 is not running
Launch VM1 with V2 fails
Launch VM1' fails
Launch VM1' succeeds, but with wrong version
API call fails
Stuck at a certain state
Launch VM1' succeeds
VM1' doesn't use V2
n times of retry fail
First call of launch VM API fails
n call of launch VM API fails
Pending
Timeout
ASG is using LC'
LC' uses wrong value
Insufficient instance capability
Security group does not exist
ELB does not exist
The AMI is not permitted to launch
AMI not available
AWS internal failure
Wrong security group
Wrong key pair
Wrong AMI
Wrong instance type
Resource unavailable
Config error
Resource unavailable
Config error
AMI does not exist
AMI is pending

Monitoring

- One of the problems with using thresholds for alerts or alarms is the number of false positives if the thresholds are set low
- Relaxing the thresholds raises the possibilities of false negatives—namely, missing actual errors
- Normal practice is to adjust the thresholds to achieve a tolerable number of false positives

Summary

- Viewing operations as a process allows us to create a process model from log lines and to use that process model to detect and sometimes repair errors caused by operational reasons

For Further Reading

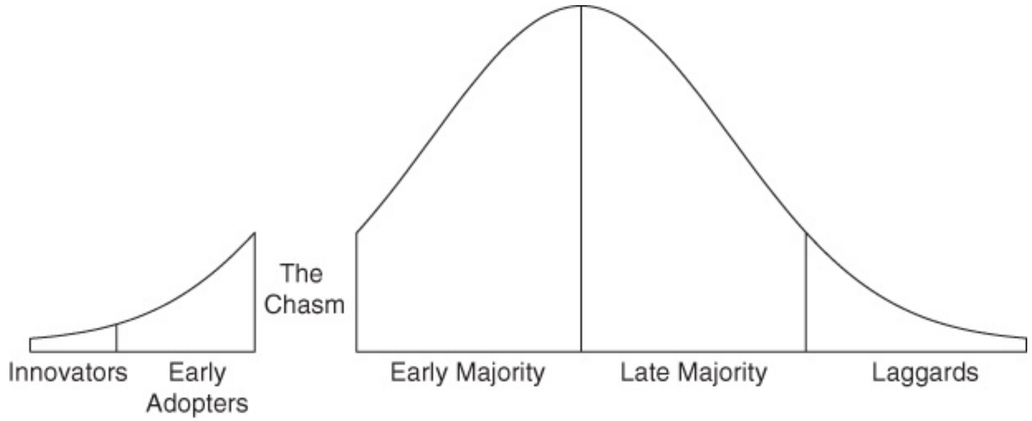
- You can find more about process mining in van der Aalst's book: Process Mining: Discovery, Conformance and Enhancement of Business Processes. Springer, 2011
- Research in error detection, diagnosis, and recovery from SEI
  - (Xu et al. 2014), (Weber et al. 2015) and website
- AWS: Error Codes—Amazon Elastic Compute Cloud
- Logstash: http://logstash.net
- Asgard: https://github.com/Netflix/asgard

The Future of DevOps

The best thing about the future is that it comes one day at a time.
— Abraham Lincoln

Introduction

- DevOps is in the process of crossing the chasm ~> diffusion of technology
  - early adopters ~> mainstream ~> majority ~> late adopters ~> the laggards



- For DevOps, the innovators are the Googles, the Amazons, or the Netflixes. The material being created includes talks, meetups, books (including this one), new tools, LinkedIn groups, and blogs.

Organizational Issues

- We discuss three different organizational issues:
  - other groups that may be involved in DevOps-style activities,
  - ownership and reorganizations, and
  - empowerment versus control.

Other groups that may be involved in DevOps-style activities

- DevOps began as a movement to break down barriers between Dev and Ops, but there other groups that cause potential problems
  - Business initiatives: new products, discounts, test marketing, and changes in the supply chain ~> BizOps
  - Data scientists: big data is used to get business insights that drive both strategic thinking and real-time customer acquisition
  - Security: making the security process and security audits more agile is not only an organizational matter but also a matter for the regulators
  - Strategic planning: having a continuous deployment pipeline may open up other business opportunities

Ownership and Reorganizations

- Updating a microservice
  - works with all downstream services & does not have negative impacts on upstream services
  - complexity, scale, and real-time performance requirements of microservices are exacerbating the problem
- Solution? let the upstream consumers' development team have co-ownership of the testing suite
  - On the other hand, end-to-end testing in the microservice environment is expensive
- And about the age of the microservices?

Empowerment Versus Control

- On the one hand, some key decisions (such as releases and A/B testing) are delegated to small teams and individuals
- Increases the velocity but also introduces significant risks to your production system
- Solution?
  - Place automated quality controls in the pipeline process itself

Two potential problems can be foreseen

- Processes have a tendency to be all-encompassing
  - One reaction to a problem is to add new elements to the process
  - Over time, processes lose their original motivation and no one recalls exactly why certain elements are in the process
- Ownership of the process
  - Development teams can own the process to deployment, or the organization can own the process
  - The ownership of the process may pass to a team unfamiliar with the process or its rationale

Process Issues

- Some of the process issues that will arise in the future are old ones such as concern about vendor lock-in
- Some are due to the use of the cloud as a platform such as charging models
- Some are due to the increased velocity of changes that are being deployed.

Vendor Lock-in and Standards

- This is not a new problem, but one that has existed in the computer industry for at least 50 years!
- One solution to the problem of vendor lock-in is the use of standards
  - it does not guarantee portability, it does simplify the problem
- Lacking standards:
  - Defensive programming: support modifiability
  - Migration programs: move programs from one tool to another
  - Provide open APIs (i.e. Jenkins)

Charging Models

1. Consumption-based. Pay for the resources that you use based on a preannounced schedule.
2. Subscription-based. Pay for unlimited or capped usage during a particular time such as a month.
3. Advertising-based. Allow advertising to appear on your web pages or displays for price reductions.
4. Market-based. Pay based on supply and demand at a particular time. Auctions are used to allocate resources to consumers

Velocity of Changes

- The qualities of concern with a deployment will change
  - Deploy form 1 time a month to 10 times a day!
- Automatic error detection/recovery is important
  - Tools to monitor and detect errors and to roll back automatically become important
- Workload and application behavior changes
  - Adjusting the threshold for new versions is a major challenge
- Environment changes
  - Monitoring tools need to become deployment-aware and aware of other process interferences

Technology Issues

- Continuous delivery pipeline will begin to be viewed as a single entity rather than as a collection of individual tools
  - There is little traceability throughout the full pipeline
  - The lack of traceability makes error diagnosis very difficult
  - Security credentials often need to be passed from one tool to another, which causes security risks
  - Many existing popular tools were created before the era of continuous delivery and deployment

Continuous Deployment Pipeline Concepts

- Environment
  - Typically contains a load balancer, the system operating in that environment, and necessary support entities such as a database and configuration parameters

Continuous Deployment Pipeline Concepts

- Deployment
  - Blue/green (or red/black) deployment: specifying the image to be deployed and the target of the deployment (number of VMs, autoscaling rules, location of the VMs)
    a) ensuring the new version is properly installed and
    b) shifting traffic to the new version when appropriate

Continuous Deployment Pipeline Concepts

- Deployment (continuing)
  - Rolling upgrade
    - Specifying the image to be deployed, the instances to be replaced, and the granularity of the rolling upgrade
  - Rollback
  - Canary deployment
    - Specifying the number of canaries and the criteria for their placement (random, customer-based, geographically based)
  - A/B testing

Continuous Deployment Pipeline Concepts

- Deployment (continuing)
  - Feature toggles
    - activation or deactivation of a feature
  - Teardown
    - An environment is torn down by deleting its VMs and other resources and removing it from any Domain Name System (DNS) entries

Continuous Deployment Pipeline Concepts

- Monitoring
  - Both the performance of the pipeline as well as the behavior of the application being deployed in the staging and production environments
- Replicating data or versions
- Service level agreements (SLAs) for the pipeline
- SLAs for the applications
- Configuration management database (CMDB)

Achieving Quality in a Continuous Deployment Pipeline

1. Don’t do anything stupid
   - You should be able to specify a series of constraints that prevent certain configurations from happening
2. Automatically detect, diagnose, and recover from errors
   - Many of these techniques should also be integrated with a more intelligent monitoring system
3. Predict completion time of operations
   - This will allow planning of deployments, where necessary, and visibility into the progress of an operation

What About Error Reporting and Repair?

- The best and most effective method to achieve normal production is to reduce the number of errors that escape the deployment process
  - Live testing such as that done by the Simian Army
  - Formal methods may finally have achieved sufficient maturity to be used in conjunction with microservices
  - A combination of static analysis together with hooks usable during execution, such as with debuggers, seems likely to enable faster repair of application errors

Final Words

- DevOps is in the middle of the chasm of adoption, but its momentum and growing number of materials will carry it into the mainstream and into normal practice

For Further Reading

- Crossing the chasm is described in a Wikipedia entry

- E. Castell. "The Present and Future of Cloud Pricing Models," IBM Cloud Products and Services, June 12, 2013