

Netflix open sources their continuous delivery platform, spinnaker

Netflix recently announced the open source release of their new continuous delivery platform.

For those of who have way too much spent time trying to piecemeal jenkins jobs together, this is certainly worth a look.

“Spinnaker is an open source, multi-cloud continuous delivery platform for releasing software changes with high velocity and confidence.

It provides two core sets of features: *cluster management* and *deployment management*”

It is a drop in replacement for their (now legacy?) Asgard platform.

A note about their deployment pipeline:

“**Pipeline:** *Pipelines* are the key deployment management construct in Spinnaker. They are defined by a sequence of stages, along with automated triggers (optional) that kick off the pipeline, parameters that get passed to all stages in the pipeline, and can be configured to issue notifications as the pipeline executes.

- Automatic triggers can be a Jenkins job, a CRON schedule, or another pipeline. You can also manually start a pipeline. Notifications can be sent out to email, SMS or HipChat on pipeline start/complete/fail.”

More info about spinnaker can be found at:

<http://spinnaker.io/>

How to get back your git commits after somebody ate them

You do a git pull and you find that someone has undone all the changes you've made today. To restore your commits, you'll need to create a branch, checkout your commit you want to restore, and merge your branch into master (or whatever branch you want to restore to).

```
git checkout -b newBranch
git checkout 459a100 .
git diff --name-only master
git add .
git status
git commit -m 'restore commits after funky merge'
git checkout master
git pull
git merge newBranch
```

Hopefully there will be no merge conflicts. If so, you'll need to sort those out before you can push. Also, some people will tell you rebase instead. However, this works for me. YMMV.

AWS Certified Solutions

Architect - Professional exam preparation

This post, [AWS Professional Solution Architect Certification Tips](#), is one of the best I've read so far regarding preparing for the AWS Solution Architect – Professional exam. I have prepared a basic study guide for myself, which is by no means exhaustive.

1 Domain 1.0: High Availability and Business Continuity

1.1 Demonstrate ability to architect the appropriate level of availability based on stakeholder requirements

- http://media.amazonwebservices.com/AWS_Building_Fault_Tolerant_Applications.pdf
- https://media.amazonwebservices.com/architecturecenter/AWS_ac_ra_ftha_04.pdf
- <http://harish11g.blogspot.com/2012/06/aws-high-availability-outage.html>
- <https://d0.awsstatic.com/whitepapers/wordpress-best-practices-on-aws.pdf>

1.2 Demonstrate ability to implement DR for systems based on RPO and RT0

- https://media.amazonwebservices.com/AWS_Disaster_Recovery.pdf

1.3 Determine appropriate use of multi-Availability Zones vs. multi-Region architectures

- <http://cloudacademy.com/blog/aws-regions-and-availability>

[y-zones-the-simplest-explanation-you-will-ever-find-around/](#)

- <http://harish11g.blogspot.in/2012/06/aws-multi-region-high-availability.html>

1.4 Demonstrate ability to implement self-healing capabilities
Content may include the following: High Availability vs. Fault Tolerance

- <http://jineshvaria.s3.amazonaws.com/public/cloudbestpractices-jvaria.pdf>
- http://media.amazonwebservices.com/AWS_Operational_Checklists.pdf (Application HA/Resilience, pg 12)

2 Domain 2.0: Costing

2.1 Demonstrate ability to make architectural decisions that minimize and optimize infrastructure cost

- <http://www.slideshare.net/AmazonWebServices/journey-through-the-cloud-cost-optimisation>
- <https://aws.amazon.com/premiumsupport/trustedadvisor/best-practices/#cost-optimizing>

2.2 Apply the appropriate AWS account and billing set-up options based on scenario

- <http://docs.aws.amazon.com/awsaccountbilling/latest/aboutv2/billing-what-is.html>
- <https://aws.amazon.com/billing/faqs/>

2.3 Ability to compare and contrast the cost implications of different architectures

- http://media.amazonwebservices.com/AWS_TCO_Web_Applications.pdf
- http://media.amazonwebservices.com/AWS_Cloud_Architectures.pdf

3 Domain 3.0: Deployment Management

3.1 Ability to manage the lifecycle of an application on AWS

- <http://www.slideshare.net/AmazonWebServices/managing-your-application-lifecycle-on-aws-continuous-integration-and-deployment>

3.2 Demonstrate ability to implement the right architecture for development, testing, and staging environments

- <https://aws.amazon.com/application-management/>
- http://d0.awsstatic.com/whitepapers/AWS_DevOps.pdf

3.3 Position and select most appropriate AWS deployment mechanism based on scenario

- <https://d0.awsstatic.com/whitepapers/overview-of-deployment-options-on-aws.pdf>

4 Domain 4.0: Network Design for a complex large scale deployment

4.1 Demonstrate ability to design and implement networking features of AWS

- <http://cloudacademy.com/blog/aws-networking-the-differences/>
- http://docs.aws.amazon.com/AmazonVPC/latest/UserGuide/VPC_Networking.html

- <http://www.slideshare.net/AmazonWebServices/sdd419-amazon-ec2-networking-deep-dive-and-best-practices-aws-reinvent-2014>

4.2 Demonstrate ability to design and implement connectivity features of AWS

- http://media.amazonwebservices.com/AWS_Amazon_VPC_Connectivity_Options.pdf

5 Domain 5.0: Data Storage for a complex large scale deployment

5.1 Demonstrate ability to make architectural trade off decisions involving storage options

- http://media.amazonwebservices.com/AWS_Storage_Options.pdf

5.2 Demonstrate ability to make architectural trade off decisions involving database options

- https://aws.amazon.com/running_databases/

5.3 Demonstrate ability to implement the most appropriate data storage architecture

- http://docs.aws.amazon.com/AmazonRDS/latest/UserGuide/CHAP_Storage.html

5.4 Determine use of synchronous versus asynchronous replication

- <https://aws.amazon.com/rds/faqs/>

6 Domain 6.0: Security

6.1 Design information security management systems and compliance controls

- <https://d0.awsstatic.com/whitepapers/Security/AWS%20Security%20Whitepaper.pdf>

6.2 Design security controls with the AWS shared responsibility model and global infrastructure

- <https://d0.awsstatic.com/whitepapers/aws-security-best-practices.pdf>

6.3 Design identity and access management controls

- <http://docs.aws.amazon.com/IAM/latest/UserGuide/best-practices.html>
- <http://docs.aws.amazon.com/IAM/latest/UserGuide/access.html>

6.4 Design protection of Data at Rest controls

- <https://aws.amazon.com/blogs/aws/new-encryption-options-for-amazon-rds/>
- https://d0.awsstatic.com/whitepapers/AWS_Securing_Data_at_Rest_with_Encryption.pdf

6.5 Design protection of Data in Flight and Network Perimeter controls

- [How-to-Address-the-PCI-DSS-Requirements-for-Data-Encryption-in-Transit-Using-Amazon-VPC](#)

7 Domain 7.0: Scalability and

Elasticity

7.1 Demonstrate the ability to design a loosely coupled system

- <https://aws.amazon.com/blogs/aws/on-using-aws-in/>
- <http://blog.bwhaley.com/loosely-coupled>
- <http://blog.bwhaley.com/loosely-coupled-part-2>

7.2 Demonstrate ability to implement the most appropriate front-end scaling architecture

- DNS
 - <https://aws.amazon.com/route53/details/>
 - <https://aws.amazon.com/route53/faqs/>
- CDN
 - <https://aws.amazon.com/cloudfront/details/>
 - <https://aws.amazon.com/cloudfront/faqs/>
- ELB
 - <https://aws.amazon.com/elasticloadbalancing/details/>
 - <https://aws.amazon.com/ec2/faqs/#elastic-load-balancing>
- Autoscaling
 - <https://aws.amazon.com/autoscaling/details/>
 - <https://aws.amazon.com/ec2/faqs/#auto-scaling>
- Containers
 - <https://aws.amazon.com/ecs/details/>
 - <https://aws.amazon.com/ecs/faqs/>
- ElasticBeanstalk
 - <https://aws.amazon.com/elasticbeanstalk/details/>
 - <https://aws.amazon.com/elasticbeanstalk/faqs/>

7.3 Demonstrate ability to implement the most appropriate middle-tier scaling architecture

- Queuing
 - <https://aws.amazon.com/sqs/details/>
 - <https://aws.amazon.com/sqs/faqs/>

- Appstream
 - <https://aws.amazon.com/appstream/>
- Email
 - <https://aws.amazon.com/ses/details/>
- Notification
 - <https://aws.amazon.com/sns/details/>
- Search
 - <https://aws.amazon.com/cloudsearch/>
 - <https://aws.amazon.com/cloudsearch/faqs/>

7.4 Demonstrate ability to implement the most appropriate data storage scaling architecture

- Relational DB
 - <https://aws.amazon.com/rds/details/>
 - <https://aws.amazon.com/rds/faqs/#hardware-scaling>
- NoSQL
 - <https://aws.amazon.com/dynamodb/details/>
 - https://aws.amazon.com/dynamodb/faqs/#scale_anchor
- Caching
 - <https://aws.amazon.com/elasticache/details/>
 - <https://aws.amazon.com/elasticache/faqs/>
- Data Warehousing
 - <https://aws.amazon.com/redshift/>
 - <https://aws.amazon.com/redshift/faqs/#scalability>
- Big Data
 - <https://aws.amazon.com/elasticmapreduce/>
 - <https://aws.amazon.com/elasticmapreduce/details/>

7.5 Determine trade-offs between vertical and horizontal scaling

- https://en.wikipedia.org/wiki/Scalability#Horizontal_and_vertical_scaling

8 Domain 8.0: Cloud Migration and Hybrid Architecture

8.1 Plan and execute for applications migrations

- <http://media.amazonwebservices.com/CloudMigration-main.pdf>

8.2 Demonstrate ability to design hybrid cloud architectures

- <https://aws.amazon.com/enterprise/hybrid/>
-