

Moi Programming

Preview

Friday, 22 February 2019

Industry Practice and Tools 1

What is the need for VCS?

Version control systems are a category of software tools that help a software team to manage changes to source code over time. These systems automatically maintain character level changes for all files stored within allowing for a complete retrace of all versions of each file. Version control software keeps track of every modification to the source in a special kind of database.

Differentiate the three models of VCSs, stating their pros and cons

01 local control system

- only one user work on a given file at a time.
- no conflicts occur.
- user wait each other for the locked file.
- works for small development team only.
- pessimistic concurrency control.
- lock-modify-unlock is rarely used.
- ex: SVN , GIT

02 centralized version control system

- users make parallel changes to their own working copies.
- conflicts are possible when multiple user edit the same file.
- optimistic concurrency control.
- ex: TFS

03 distributed version control

- users work in their own repository.
- from time to time, the local repository is pushed to the central repository.
- ex: Mercurial

Git and GitHub, are they same or different? Discuss with facts.

- GIT--
- installed locally
 - maintain by the linux foundation
 - focused on version control and code sharing
 - primarily a command-line tool
 - no user management features
 - open source licensed

--GITHUB--

- hosted in the cloud
- purchased in 2018 by microsoft
- focused on centralized source code hosting
- administered through the web
- built-in user management
- includes the free tier and pay for use

Compare and contrast the Git commands, commit and push

Git Commands; Commit and Push

Git Commit records changes to the repository while Git Push updates remote refs along with associated objects.
Basically Commit is connected with our local repository while Push interacts with the remote repository

pushing

{git push [remote name] [local name]}

committing

About Me

Moi Programming
View my complete profile

Blog Archive

February 2019 (2)

Labels

Report Abuse

Search This Blog

Search

Home

Industry Practice and Tools

Popular Posts

Programming Applications and Frameworks
Compare and contrast declarative and imperative paradigms

Declarative Programming

```
{git commit -m "[your message here]"}
```

Discuss the use of staging area and Git directory

- staging helps you split up one large change into multiple commits
- staging helps in reviewing changes
- staging helps when a merge has conflicts
- staging helps you keep extra local files hanging around
- staging helps you sneak in small changes

- supposedly containing files and data to run.

Explain the collaboration workflow of Git, with example

A Git Workflow is a recipe or recommendation for how to use Git to accomplish work in a consistent and productive manner. Git workflows encourage users to leverage Git effectively and consistently. Git offers a lot of flexibility in how users manage changes. Given Git's focus on flexibility, there is no standardized process on how to interact with Git. When working with a team on a Git managed project, it's important to make sure the team is all in agreement on how the flow of changes will be applied. To ensure the team is on the same page, an agreed upon Git workflow should be developed or selected. There are several publicized Git workflows that may be a good fit for your team. Here, we'll be discussing some of these workflow options

Discuss the benefits of CDNs

If you don't know much about Content Delivery Networks, or CDNs, then this Learning Center is the place to start. If you already know the basics, then feel free to check out the other links on this page to find another topic. Keep checking back as we continue to add new content every month.

CDNs carry a significant portion of the world's Internet traffic. They are ubiquitous in presence and mitigate the toughest challenges of delivering content over the Internet. But why are CDNs so pervasive? Why is it that everyone, from small and medium content providers, to the world's large corporations rely on CDNs to provide a seamless web experience to their end users?

CDNs became an essential tool to successfully conduct business online for one main reason: the Internet was not originally architected to do all of the amazing things that it does today! It simply wasn't built to handle the demands of massive amount of data, live high definition video, flash sales, and large downloads that people expect today. CDNs were specifically built to make the Internet work better, deliver media at scale, and to enable all of the connected experiences you can imagine.

In specific terms, CDN technology should provide the following primary benefits to a business:- Performance

- Availability
- Security
- Intelligence

How CDNs differ from web hosting servers?

Web Hosting is used to host your website on a server and let users access it over the internet. A content delivery network is about speeding up the access/delivery of your website's assets to those users.

Traditional web hosting would deliver 100% of your content to the user. If they are located across the world, the user still must wait for the data to be retrieved from where your web server is located. A CDN takes a majority of your static and dynamic content and serves it from across the globe, decreasing download times. Most times, the closer the CDN server is to the web visitor, the faster assets will load for them.

Web Hosting normally refers to one server. A content delivery network refers to a global network of edge servers which distributes your content from a multi-host environment.

Identify free and commercial CDNs

A content delivery network, commonly referred to as CDN, is a network of edge servers strategically placed across the globe with the purpose of delivering digital content to users as fast as possible. When a user makes a request it is routed to the nearest CDN edge server, which significantly reduces the latency. A CDN allows all users, no matter the geographical location, to have fast loading content for an unquestionably improved experience.

- free
- CloudFlare
 - Incapsula
 - Photon by Jetpack
 - Swarmify

commercial
 -akamai
 -amazon
 -CDNetworks
 -CDN77

Discuss the requirements for virtualization

-Processor that supports Intel VT-X

-Minimum 2 GB Memory (NAS reserves 1.5 GB Memory): TS-x51 series

-Minimum 4 GB Memory (NAS reserves 2 GB Memory): TS-x70, TS-x70 Pro, TS-ECx80 Pro, TS-x70U-RP, TS-x79U-RP, TS-ECx79U-RP, TS-ECx79U-SAS-RP, SS-ECx79U-SAS-RP and TS-ECx80U-RP series

-Minimum 550 MB Hard disk space

-Minimum two Ethernet cables

-Supported NAS series: TS-x51/TS-x51-4G, TS-x70, TS-x70 Pro, TS-x70U-RP, TS-x79U-RP, TS-ECx79U-RP, TS-ECx79U-SAS-RP, SS-ECx79U-SAS-RP, TS-ECx80 Pro and TS-ECx80U-RP, series

Discuss and compare the pros and cons of different virtualization techniques in different levels

Hardware Virtualization – This is the use of computing hardware in an environment separate from the actual existence of the hardware. There are different levels of hardware virtualization including full, partial and para virtualization. Each has its own advantages and unique characteristics.

Virtual Machine – A virtual machine is a software clone of a real computer that executes programs as if it were the computer. Other types of software virtualization techniques include virtual appliance, application virtualization, cross-platform virtualization and OS virtualization.

Storage Virtualization – Just as the name implies, the focus is on separating physical storage from actual storage.

Desktop Virtualization – The ability to operate a computing environment remotely.

Network Virtualization – The creation of work space within a larger network or across networks using virtualization techniques.

Identify popular implementations and available tools for each level of visualization

Hardware Virtualization : CPU ,memory ,network I/O capacity

Virtual Machine :RV tools,PowerShell,Vcontrol

Storage Virtualization :Virtual storage appliances,vSphere Storage Appliance,HP LeftHand Virtual SAN Appliance

Desktop Virtualization :Amazon workspace,IBM cloud,Cisco VXi

Network Virtualization :Nagios XI,Cacti,Zenmap

What is the hypervisor and what is the role of it?

A hypervisor is a process that separates a computer's operating system and applications from the underlying physical hardware.

hypervisor plays an important role in the virtualization scenario by virtualization of hardware.it provides support for running multiple operating system concurrently in virtual servers created within a physical server.

the virtualization layer is the software responsible for hosting and managing all VMs.the virtualization layer is a hypervisor running directly on the hardware.

ex:VMWare

How does the emulation is different from VMs?

Emulation, in short, involves making one system imitate another. For example, if a piece of software runs on system A and not on system B, we make system B "emulate" the working of system A. The software then runs on an emulation of system A.

In this same example, virtualization would involve taking system A and splitting it into two servers, B and C. Both of these "virtual" servers are independent software containers, having their own access to software based resources – CPU, RAM, storage and networking – and can

be rebooted independently. They behave exactly like real hardware, and an application or another computer would not be able to tell the difference.

Compare and contrast the VMs and containers/dockers, indicating their advantages and disadvantages

VMs

- Heavyweight
- Limited performance
- Each VM runs in its own OS
- Hardware-level virtualization
- Startup time in minutes
- Allocates required memory
- Fully isolated and hence more secure

Containers

- Lightweight
- Native performance
- All containers share the host OS
- OS virtualization
- Startup time in milliseconds
- Requires less memory space
- Process-level isolation, possibly less secure

on February 22, 2019



No comments:

Post a Comment

Enter your comment...



Comment as:

Moi Programm ▼

Sign out

Publish

Preview

☐ Notify me

Home

Subscribe to: [Post Comments \(Atom\)](#)