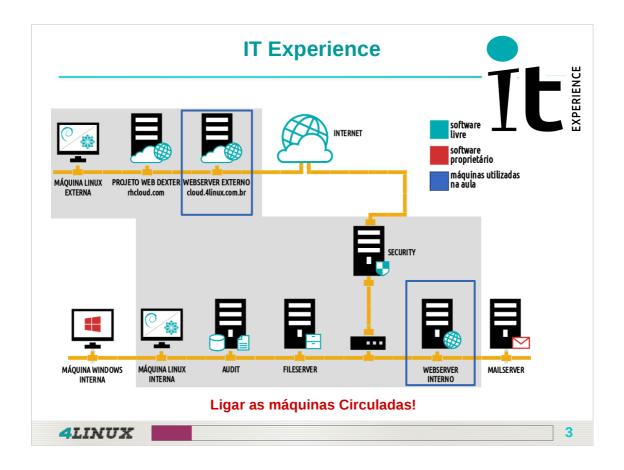


Curso 4451

Linux Security Administration in Cloud



Anotações:		



Anotações:		

Objetivos da Aula

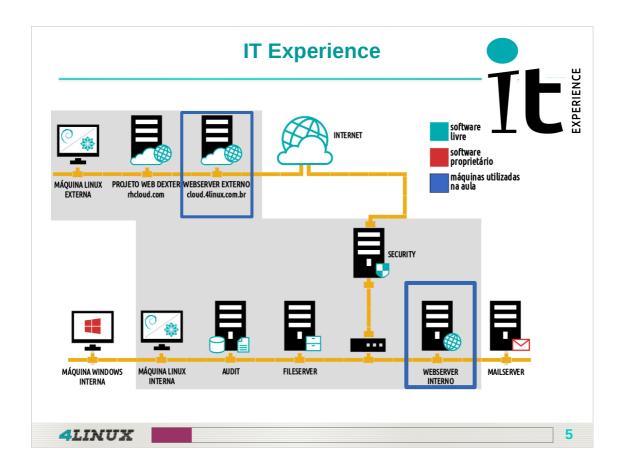
Aula 09

- > Entender o conceito de capacity planning;
- Utilizar ferramentas de monitoramento e análise nativas de um sistema GNU/Linux;
- Utilizar o pacote sysstat para monitoramento e análise dos servidores da empresa Dexter.



4LINUX

Anotações:		



Anota	ções:			

O que é Capacity Planning?

Definição Básica:

- "Capacity Planning" é o conjunto de ações/processos utilizados para determinar a capacidade de produção de um determinado sistema;
- Na área de T.I., o foco principal do "Capacity Planning" é garantir o alto nível e a alta performance dos serviços e soluções entregues à empresa, atendendo completamente as suas necessidades.

4LINUX

6

Planejamento de Capacidade

O processo de gerenciamento da capacidade é particularmente importante em empresas com infraestrutura de grande porte, imagine por exemplo que nossa empresa Dexter possui cerca de 300 mil colaboradores, e que você é o responsável pelo planejamento e construção da plataforma de e-mail a ser utilizada na empresa, Quais seriam as dimensões da infraestrutura para suportar esta plataforma? Qual seria o ônus de uma indisponibilidade de digamos... 15 minutos?

Arquitetar e dimensionar corretamente a infra necessária para cenários como o descrito acima com certeza configura um grande desafio de gerenciamento de TI, em um projeto com tal nível de complexidade o profissional deverá prever possíveis falhas e qualquer tipo de surpresa a fim de que possa evitá-las mantendo a infra alinhada às expectativas e demandas do negócio.



Planejamento de Capacidade

Ao planejar e prover os recursos de uma infraestrutura é importante ter em mente que o Planejamento Capacidade deve considerar não só o cenário atual, mas também possíveis expansões de negócio como crescimento natural do negócio ou demandas pontuais que exijam mais dos sistemas.

Imagine se em um projeto de expansão nosso sistema de e-mail passa de 300 para 500 mil caixas (Contas de Usuário) Com certeza um aumento de demanda neste nível poderia causar problemas em um sistema mal arquitetado.

Utilizar Técnicas de "Capacity Planning" para definir métricas, que permitam aplicar da melhor forma possível os recursos da empresa, também é uma prática DevOps! > Quais serviços estão rodando no servidor? > Como está o consumo de memória no servidor? > Como está o processamento no servidor? > O servidor possui o hardware ideal para as aplicações as quais foi designado?

Mapeamento do Sistema

Existem várias questões a serem analisadas em processos de Planejamento de Capacidade, Em nosso cenário analisaremos o desempenho e capacidade computacional dos computadores da empresa Dexter de forma a obter uma visão mais completa do parque de servidores da organização.

Capacity Planning

Servidor: WebServer Interno

Análise de Capacidade com Comandos Internos:

Comandos nativos do sistema GNU/Linux podem ser utilizados para obter informações importantes sobre o desempenho do sistema.

- 1 Verifique a quantidade de processadores:
- 1# cd /proc
- 2# cat cpuinfo
- 2 Verifique a quantidade de memória disponível:
- 3# cat meminfo
- 4# free -m

4LINUX			9
--------	--	--	---

Anotações:		

Capacity Planning

Servidor: WebServer Interno

Análise de Capacidade com Comandos Internos:

Outro tipo de verificação importante é em relação a quanto cada processo/aplicação requer de uso da CPU.

1 – Execute o comando ps e verifque quais processos estão abertos:

1# ps aux

2 – Para definir quais dos processos listados estão utilizando mais recursos, é possível utilizar o comando top:

2# top

4LINUX 10

Anotações:	

Capacity Planning Servidor: WebServer Interno Load Average: A primeira linha do comando top também pode ser observada com o comando up-time: # uptime Esse comando exibe o horário, seguido do tempo que o sistema está ligado (up) e do número de usuários logados (users), mas a informação importante relativa a "capacity planning" trazida por este comando é o "load average".

Como funciona o "load average"?

4LINUX

O "load average" é a média de carga de processamento há 1 minuto atrás, há 5 minutos atrás e há 15 minutos atrás;

- O valor de "load average" exibido deve ser inferior a quantidade de processadores disponíveis no servidor;
- Para um hardware com dois processadores um valor ideal de load average será qualquer coisa menor ou igual a 2;
- Qualquer valor superior a isso indicaria "gargalo", ou seja, fila de processamento no sistema.

Capacity Planning

Servidor: WebServer Interno

Verificações de Memória com vmstat:

O comando vmstat é uma ferramenta de capacity planning capaz de gerar alguns tipos de resumos, como resumos de memória ou armazenamento.

- 1 Crie um relatório com informações sobre memória:
- 1# vmstat -s
- 2 É possível executar o vmstat com uma taxa de atualização específica e com quantidade de linhas específicas na saída:
- 2# vmstat 1 10

4LINUX		1	2

Anotações:	

Capacity Planning com sysstat

- O sysstat é um pacote que contém ferramentas utilizadas para análise de performance do sistema, composto pelos seguintes itens:
 - ➤ lostat → Reporta utilização de CPU e estatísticas de I/O do disco;
 - ➤ mpstat → Reporta estatísticas globais e por processador;
 - > sar → Recolhe e reporta informações de atividade do sistema;

13

> sadf → Visualiza os dados colhidos pelo comando sar.

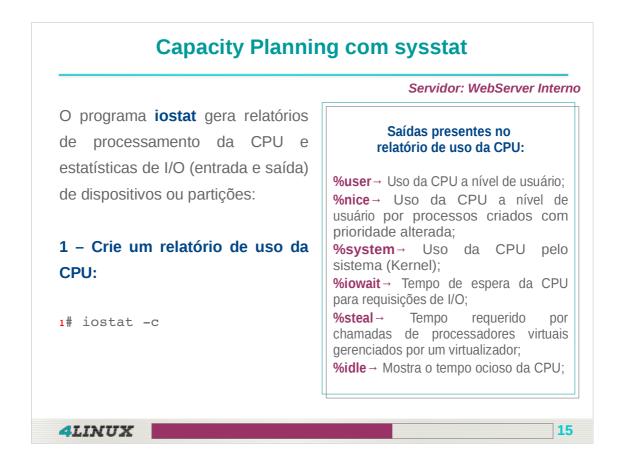
4LINUX



•			
•			

Capacity Planning com sysstat Servidor: WebServer Interno/ WebServerCloud 1 - Verifique se o pacote sysstat está instalado: 1# rpm -qa | grep sysstat 2# dpkg -L | grep sysstat 2 - Caso não esteja, instale-o através dos seguintes comandos: 1# yum install sysstat 4# apt-get install sysstat

Anotações:	



NOTA:

O comando iostat gera dois tipos de relatórios, sendo o primeiro um relatório de uso da CPU, com as informações listadas na coluna direita do slide, e o segundo um relatório de disco conforme exemplo que segue no próximo slide.

Capacity Planning com sysstat

Servidor: WebServer Interno

- 1 Crie um relatório de uso de Disco:
- 1# iostat -d
- 2 Utilize o comando abaixo para gerar um arquivo que ocupe algum espaço em disco:
- 2# cd /backup
- 3# tar -cf /backup/bkp_etc.tar
 /etc

Saídas presentes no Relatório de disco:

tps→ Indica o número de transferência por segundo realizadas por cada disco;

Blk_read / s→ Cria uma métrica baseada em um determinado número de blocos, com a quantidade de dados lidos por segundo;

%Blk-wrtn / s → Cria uma métrica baseada em um determinado número de blocos, com a quantidade de dados gravados por segundo;

%Blk_read → Total de blocos lidos;

%Blk_wrtn → Total de blocos escritos;

4	T.	Ŧ	ħ	f	Ŧ	T	3	7
_	_	_		•	•	20	- 1	-

Anotações:		

Capacity Planning com sysstat Servidor: WebServer Interno 3 - Utilize o comando iostat para gerar um novo relatório de disco e verifique a taxa de gravação (Blk_wrtn/s) no disco de backup (sdb): 1# iostat -d 4 - Verifique informações gerais de processamento: 2# mpstat 5 - Faça verificações em um único processador e em um único processo: 3# mpstat -P 0 4# pidstat \$(pgrep cron)

mpstat e pidstat

O programa **mpstat** trabalha de forma similar ao iostat, porém reporta informações apenas sobre o processador e permite um nível maior de detalhamento em relação as informações reportadas;

Já o comando **pidstat** pode ser extremamente útil para monitorar as requisiçõpes feitas por aplicações especificas através de seu PID.



Sar (System Activity Reporter)

O programa **sar** provê informações semelhantes as informações obtidas pelos comandos vmstat, iostat e netstat porém com a capacidade de aglutinar essas informações em **relatórios** gerados regularmente.

NOTA:

No CentOS a criação de relatórios pelo sar é habilitada no processo de instalação do pacote sysstat.

Capacity Planning com sysstat

Servidor: WebServerCloud

Através do comando **sar** é possível gerar estatísticas com exibições em intervalos de tempos regulares e em quantidades pré definidas:

- 1 Verifique as estatísticas de CPU com um intervalo de 5 segundos e 3 repetições:
- 1# sar -u 5 3
- 2 Verifique estatísticas de I/O com os mesmos intervalos usados acima:
- 2# sar -b 5 3
- 3 Crie um relatório completo do servidor
- 3# sar -A > relatoriodexter.txt
- 4# cat relatoriodexter.txt

Principais Opções do sar:

- -b→ Atividades do buffer (Estatísticas de Transferência);
- -d → Dispositivos de bloco;
- -w→ Volume de troca e SWAP;
- -u → Utilização da CPU;
- -A → Todos os relatórios.

4LINUX

Anotações:			
	 1 1	1	

Laboratório Dexter Servidor: WebServerInterno/ WebServerCloud Stress Testing: Formato de Capacity Planning onde um determinado servidor ou serviço é testado em relação a sua capacidade operacional; A partir de um teste de stress pode-se definir pontos de ruptura, especificações de hardware mínimo, previsão de falhas, etc; Em nossos cenário executaremos um teste de stress nos servidores Cloud e WebServerInterno; Durante os testes utilizaremos o projeto stress disponível em: http://people.seas.harvard.edu/~apw/stress/

Objetivos dos testes de Planejamento de Capacidade:

4LINUX

Observar o comportamento do sistema em determinadas situações ou ao executar determinadas aplicações;

- Estimar o impacto de novas aplicações ou modificações no funcionamento do sistema e nas aplicações em execução no servidor;
- Planejar antecipadamente questões relacionadas a desempenho e demanda diante de uma determinada aplicação;
- > Se possível obter economia de custos, serviços de Cloud Computing possui seus valores baseados no hardware alocado ao cliente.

Laboratório Dexter



Testes de Stress na CPU:

- 1 Verifique as métricas atuais de processamento:
- 1# iostat -c ; uptime
- 3 No terminal execute o seguinte comando:
- 2# stress --cpu 2k --timeout 30
- 3 Em sequida verifique novamente as métricas:
- 3# iostat -c ; uptime

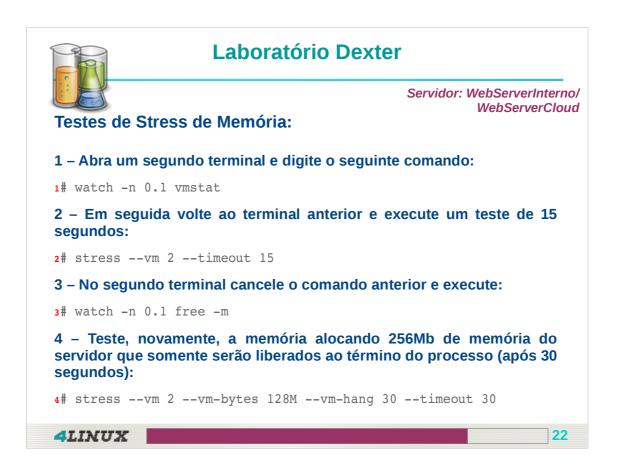
Servidor: WebServerInterno/ WebServerCloud

Principais opções do stress:

- **--cpu** → Cria um fork com a quantidade de processos especificados;
- --hdd → Gera chamadas de sistemas através de funções write(), simulando escrita no hd;
- --io → Gera chamadas de sistemas através de funções sync();
- --vm→ Cria um fork com a quantidade de processos especificados alocando memória da CPU;
- **--vm-bytes** → Especifica quanto de memória será alocada, utilizado após a opção **vm**.

4LINUX

Anotações:	



O Jmeter pode ser uma boa opção

Saindo do escopo de Planejar a infraestrutura e indo direto ao Planejamento de Capacidade relacionado ao funcionamento e desempenho de serviços o uso da plataforma JMeter torna-se uma interessante alternativa.

Jmeter é uma aplicação apache open source, projetada para efetuar testes medindo o desempenho/performance de aplicações web.

Laboratório Dexter

Servidor: WebServerInterno/ WebServerCloud

Testes de Stress de Disco e I/O:

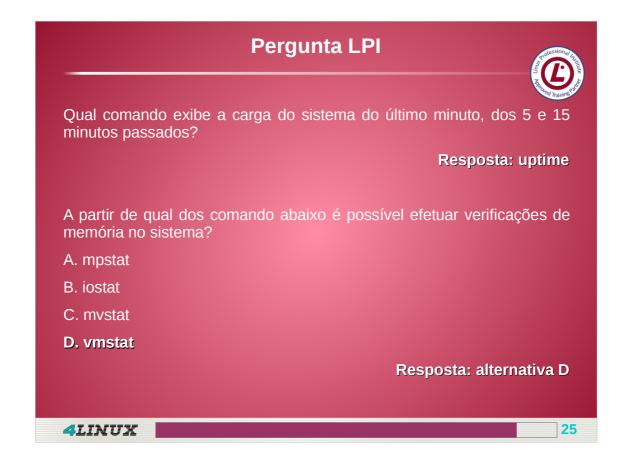
- 1 Volte ao segundo terminal e monitore o disco:
- 1# watch -n 0.1 iostat -d
- 2 Em seguida, execute no terminal original:
- 2# stress --io 6 --timeout 10s
- 3# stress --hdd 20
- 3 Também é possível executar múltiplos testes, simultaneamente:
- 4# stress --cpu 1k --vm 2 --vm-bytes 128M --timeout 20

4LINUX		2	3
-LINCOA			~

Anotações:			
		 , , , , , ,	

Pergunta LPI Qual comando exibe a carga do sistema do último minuto, dos 5 e 15 minutos passados? A partir de qual dos comando abaixo é possível efetuar verificações de memória no sistema? A. mpstat B. iostat C. mvstat D. vmstat

Anotações:		



RESPOSTA CORRETA: uptime

Conforme verificado em aula o comando **uptime** reúne informações a sobre o processamento do sistema, a partir dele é possível verificar a quanto tempo o sistema está em funcionamento, quantos usuários estão conectados no momento e as médias de carga do sistema para 1, 5, e 15 minutos atrás.

Alternativa D: RESPOSTA CORRETA!

O comando vmstat é utilizado para reportar estatísticas sobre a memória disponível no sistema.

Próximos Passos

Para que você tenha um melhor aproveitamento do curso, participes das seguintes atividades disponíveis no Netclass:

- > Executar as tarefas do **Practice Lab**;
- Resolver o **Desafio Appliance Lab** e postar o resultado no Fórum Temático;
- Responder as questões do Teste de Conhecimento sobre o conteúdo visto em aula.

Mãos à obra!

4LINUX

