

MÓDULO GERENCIAMENTO DE INFRAESTRUTURA DE NUVEM

Curso Técnico em Redes / Infraestrutura

Ambiente das práticas: VPS Linux Debian + acesso remoto via ttyd no Chromebook

Carga horária: 40h

Versão: 2026

Professor: Marcos Brandão

SUMÁRIO

I UNIDADE — FUNDAMENTOS E CONTAINERS

1. Introdução à Computação em Nuvem
 - 1.1 O que é computação em nuvem
 - 1.2 Modelos de serviço (IaaS, PaaS, SaaS)
 - 1.3 Vantagens da nuvem
 - 1.4 Cenários reais de uso
2. Nuvem Pública e Privada
 - 2.1 Conceitos
 - 2.2 Diferenças
 - 2.3 Casos de uso
 - 2.4 VPS e infraestrutura virtual
3. Ambiente Prático da Disciplina
 - 3.1 VPS Linux Debian
 - 3.2 Acesso remoto via ttyd
 - 3.3 Estrutura do laboratório
 - 3.4 Organização dos projetos
4. Linux para Infraestrutura de Nuvem
 - 4.1 Gerenciamento de usuários
 - 4.2 Serviços Linux
 - 4.3 Monitoramento básico
 - 4.4 Rede e portas
5. Containers e Docker
 - 5.1 Conceito de containers
 - 5.2 Containers x máquinas virtuais
 - 5.3 Instalação do Docker

- 5.4 Criação e gerenciamento de containers
 - 5.5 Redes Docker
 - 5.6 Volumes Docker
-

II UNIDADE — ORQUESTRAÇÃO E MONITORAMENTO

- 6. Monitoramento de Infraestrutura
 - 6.1 Conceitos de monitoramento
 - 6.2 Métricas importantes
 - 6.3 Logs e análise
 - 7. Zabbix e Nagios
 - 7.1 Instalação
 - 7.2 Monitoramento de hosts
 - 7.3 Alertas
 - 8. Prometheus e Grafana
 - 8.1 Coleta de métricas
 - 8.2 Dashboards
 - 8.3 Visualização em tempo real
 - 9. Docker Swarm
 - 9.1 Conceitos de orquestração
 - 9.2 Cluster
 - 9.3 Escalabilidade
 - 10. Kubernetes
 - 10.1 Conceitos principais
 - 10.2 Pods e deployments
 - 10.3 Serviços
 - 11. Escalonamento
 - 11.1 Escalonamento vertical
 - 11.2 Escalonamento horizontal
 - 12. Balanceamento de carga e Proxy Reverso
 - 12.1 Nginx
 - 12.2 Reverse Proxy
 - 12.3 Balanceamento
 - 13. Encaminhamento de Porta
 - 13.1 NAT
 - 13.2 Port forwarding
 - 13.3 Segurança
 - 14. Projeto Integrador
-

I UNIDADE

CAPÍTULO 1 — INTRODUÇÃO À COMPUTAÇÃO EM NUVEM

1.1 O que é Computação em Nuvem

Computação em nuvem é a entrega de serviços computacionais pela internet.

Ao invés de comprar vários computadores físicos para executar sistemas, empresas alugam recursos em servidores remotos.

Esses recursos incluem:

- processamento
 - armazenamento
 - banco de dados
 - aplicações
 - redes
 - backup
 - inteligência artificial
-

Exemplo do mundo real

Imagine uma escola que possui:

- sistema acadêmico
- site
- banco de dados
- arquivos dos alunos

Ao invés de manter tudo em um servidor físico dentro da escola, ela pode contratar uma VPS na nuvem.

Assim:

- os sistemas ficam disponíveis 24h
 - há acesso remoto
 - o custo é menor
 - a manutenção é simplificada
-

1.2 Modelos de Serviço

IaaS — Infrastructure as a Service

O provedor entrega:

- servidor virtual
- rede
- armazenamento

O administrador gerencia:

- Linux
- serviços
- aplicações

Exemplo:

- VPS Debian
-

PaaS — Platform as a Service

O provedor entrega:

- infraestrutura
- ambiente pronto

O usuário apenas publica a aplicação.

Exemplo:

- Heroku
 - Railway
-

SaaS — Software as a Service

O sistema já está pronto para uso.

Exemplo:

- Google Drive
 - Gmail
 - Microsoft 365
-

1.3 Vantagens da Nuvem

Escalabilidade

Aumentar recursos rapidamente.

Exemplo:

Um site recebe muitos acessos e aumenta CPU e RAM.

Alta disponibilidade

Serviços ficam disponíveis continuamente.

Acesso remoto

Pode ser acessado de qualquer lugar.

Redução de custos

Não há necessidade de comprar servidores físicos caros.

PRÁTICA — Conhecendo a VPS

Objetivo

Identificar informações do servidor Linux.

Comando

```
uname -a
```

Resultado esperado

Exemplo:

```
Linux servidor 6.1.0-18-amd64 x86_64 GNU/Linux
```

Explicação

Informação	Significado
Linux	Sistema operacional
servidor	Nome do host
x86_64	Arquitetura 64 bits

PRÁTICA — Verificando uso do servidor

Comando

```
uptime
```

Resultado esperado

```
22:40:11 up 5 days, 3 users, load average: 0.10, 0.20, 0.30
```

Explicação

Informação	Significado
up 5 days	tempo ligado
3 users	usuários conectados
load average	carga do servidor

CAPÍTULO 2 — NUVEM PÚBLICA E PRIVADA

2.1 Nuvem Pública

Infraestrutura compartilhada entre vários clientes.

Exemplos:

- AWS
 - Azure
 - Google Cloud
-

Características

- baixo custo inicial
 - alta escalabilidade
 - gerenciamento simplificado
-

2.2 Nuvem Privada

Infraestrutura exclusiva.

Pode estar:

- na empresa
 - em datacenter privado
-

Características

- maior controle
 - maior segurança
 - maior custo
-

Comparação

Pública	Privada
Compartilhada	Exclusiva
Mais barata	Mais cara
Fácil expansão	Controle total

PRÁTICA — Descobrimo IP público

Comando

```
curl ifconfig.me
```

Resultado esperado

```
191.xxx.xxx.xxx
```

Explicação

Mostra o IP público da VPS.

Esse IP é usado para:

- acessar aplicações
 - liberar portas
 - configurar DNS
-

CAPÍTULO 3 — AMBIENTE PRÁTICO DA DISCIPLINA

3.1 VPS Linux Debian

A VPS será o servidor principal das práticas.

Nela serão executados:

- Docker
 - Grafana
 - Nginx
 - Prometheus
 - aplicações web
-

3.2 ttyd

O ttyd permite acessar o terminal Linux pelo navegador.

Fluxo do laboratório

Chromebook



Navegador



ttyd



Terminal Linux da VPS

Vantagens

- não precisa instalar programas
 - acesso remoto fácil
 - funciona em Chromebook
-

PRÁTICA — Verificando usuários conectados

Comando

who

Resultado esperado

aluno pts/0 2026-05-24 19:30

Explicação

Mostra:

- usuário conectado
 - terminal utilizado
 - horário do login
-

CAPÍTULO 4 — LINUX PARA INFRAESTRUTURA

4.1 Gerenciamento de Serviços

Linux utiliza serviços para executar aplicações em segundo plano.

Exemplos:

- nginx
 - docker
 - ssh
 - mysql
-

systemctl

Ferramenta de gerenciamento de serviços.

PRÁTICA — Verificando serviço

Comando

```
systemctl status ssh
```

Resultado esperado

```
active (running)
```

Explicação

Indica que o SSH está funcionando.

PRÁTICA — Reiniciando serviço

Comando

```
systemctl restart nginx
```

Explicação

Reinicia o servidor web.

Muito usado após alterações de configuração.

4.2 Monitoramento básico

Comando TOP

top

Explicação

Mostra:

- CPU
 - memória
 - processos
 - carga do sistema
-

4.3 Verificando portas

Comando

ss -tulnp

Explicação

Lista:

- portas abertas
 - serviços ativos
 - processos usando rede
-

CAPÍTULO 5 — CONTAINERS E DOCKER

5.1 O que é Container

Container é um ambiente isolado para executar aplicações.

Ele compartilha o kernel do Linux.

Vantagens

- leve
 - rápido
 - portátil
 - fácil replicação
-

5.2 Docker

Docker é a principal plataforma de containers.

Fluxo Docker

Imagem → Container → Aplicação

5.3 Instalação do Docker

Comando

```
apt update  
apt install docker.io -y
```

Verificando instalação

```
docker --version
```

Resultado esperado

Docker version 24.x.x

5.4 Executando primeiro container

Comando

```
docker run hello-world
```

Explicação

O Docker:

1. baixa imagem
 2. cria container
 3. executa teste
-

Resultado esperado

Mensagem de sucesso do Docker.

5.5 Listando containers

Containers ativos

```
docker ps
```

Todos containers

```
docker ps -a
```

Explicação

Comando	Função
docker ps	ativos
docker ps -a	todos

5.6 Executando servidor web

Comando

```
docker run -d -p 8080:80 nginx
```

Explicação

Parte	Significado
-d	segundo plano
-p	mapeamento de portas
8080:80	VPS → container

Acessando

http://IP_DA_VPS:8080

II UNIDADE

CAPÍTULO 6 — MONITORAMENTO

6.1 Importância do monitoramento

Monitoramento permite:

- identificar falhas
 - detectar invasões
 - medir desempenho
 - evitar indisponibilidade
-

Métricas importantes

Métrica	Importância
CPU	processamento
RAM	memória
Disco	armazenamento
Rede	tráfego

PRÁTICA — Instalando htop

Comando

```
apt install htop -y
```

Executando

```
htop
```

Explicação

Ferramenta visual para monitoramento em tempo real.

CAPÍTULO 7 — ZABBIX

7.1 Conceito

Zabbix monitora:

- servidores
 - switches
 - aplicações
 - containers
-

Componentes

Componente	Função
Server	coleta
Agent	envia dados
Frontend	interface

PRÁTICA — Instalando agente

Comando

```
apt install zabbix-agent -y
```

Iniciando serviço

```
systemctl enable zabbix-agent  
systemctl start zabbix-agent
```

Verificando status

```
systemctl status zabbix-agent
```

CAPÍTULO 8 — PROMETHEUS E GRAFANA

Prometheus

Coleta métricas dos serviços.

Grafana

Transforma métricas em dashboards.

PRÁTICA — Grafana com Docker

Comando

```
docker run -d \  
--name grafana \  
-p 3000:3000 \  
grafana/grafana
```

Acesso

http://IP_DA_VPS:3000

Login padrão

Usuário	Senha
---------	-------

admin	admin
-------	-------

CAPÍTULO 9 — DOCKER SWARM

Conceito

Docker Swarm gerencia múltiplos containers.

Vantagens

- escalabilidade
 - alta disponibilidade
 - replicação automática
-

Inicializando cluster

Comando

```
docker swarm init
```

Verificando nós

```
docker node ls
```

Criando serviço

```
docker service create \  
--name web \  
-p 80:80 \  
nginx
```

Escalando serviço

docker service scale web=3

Explicação

Cria 3 containers automaticamente.

CAPÍTULO 10 — KUBERNETES

Conceitos

Kubernetes é um orquestrador avançado.

Componentes

Componente	Função
Pod	menor unidade
Deployment	gerencia pods
Service	exposição

Exemplo de deployment

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: nginx
spec:
  replicas: 2
```

CAPÍTULO 11 — ESCALONAMENTO

Vertical

Aumenta recursos da máquina.

Exemplo:

- 2 GB → 8 GB RAM

Horizontal

Aumenta quantidade de servidores/containers.

Exemplo:

- 1 container → 5 containers

CAPÍTULO 12 — BALANCEAMENTO E PROXY REVERSO

Proxy reverso

Recebe requisições e encaminha internamente.

Exemplo

Cliente → Nginx → Aplicação

PRÁTICA — Instalando Nginx

```
apt install nginx -y
```

Testando configuração

```
nginx -t
```

Reiniciando

```
systemctl restart nginx
```

CAPÍTULO 13 — ENCAMINHAMENTO DE PORTA

Conceito

Permite acesso externo a serviços internos.

Exemplo Docker

```
docker run -p 8080:80 nginx
```

Explicação

Porta	Função
8080	VPS
80	container

Verificando portas

ss -tulnp

CAPÍTULO 14 — PROJETO FINAL

Objetivo

Criar ambiente completo contendo:

- Docker
 - Grafana
 - Prometheus
 - Nginx
 - aplicação web
-

Etapas

1. Instalar Docker

```
apt install docker.io -y
```

2. Criar rede Docker

```
docker network create infra
```

3. Executar Grafana

```
docker run -d \  
--name grafana \  
--network infra \  
-p 3000:3000 \  
grafana/grafana
```

4. Executar Nginx

```
docker run -d \  
--name web \  
--network infra \  
-p 8080:80 \  
nginx
```

Resultado esperado

Os alunos terão:

- ambiente real de nuvem
 - containers funcionando
 - monitoramento ativo
 - acesso remoto pela VPS
-

CONCLUSÃO

Ao final da disciplina, o aluno será capaz de:

- administrar serviços Linux
- trabalhar com containers
- criar ambientes em nuvem
- monitorar infraestrutura
- implementar balanceamento
- utilizar Docker e Kubernetes
- realizar deploy de aplicações em VPS Linux