InfraMedia Manual

Sistema operacional para appliances de servidores de mídia

Flussonic

Table of contents

1. Produtos

2. InfraMedia

2.0.1 InfraMedia

Flussonic InfraMedia (doravante InfraMedia) é o sistema operacional dos appliances de servidores de mídia. É um Linux com raiz imutável, entregue como uma única imagem junto com a aplicação, mais um daemon de gestão que configura a máquina por HTTP.

Seu propósito é transformar um servidor em um aparelho acabado que funciona sem administração. Quem configura a máquina não é uma pessoa no console, e sim a aplicação pela qual o aparelho existe, através da API do configurador.

Como o sistema é construído está descrito na página de arquitetura técnica, a instalação na página de instalação, e as atualizações e a reversão na página de atualização do firmware.

Por que um sistema imutável

Os arquivos de sistema e executáveis do InfraMedia não podem ser alterados enquanto a máquina está em serviço: a raiz é composta de imagens squashfs preparadas de antemão e montada somente leitura. Todo o resto decorre disso.

- **A máquina no rack é exatamente o que foi entregue.** Um arquivo de sistema não pode ser alterado nem substituído, portanto também não existem os ajustes acumulados ao longo de anos de que ninguém se lembra. Duas máquinas da mesma versão são a mesma máquina.
- **O sistema operacional e a aplicação chegam em uma única entrega.** São construídos e testados juntos, e a junção entre eles é do fornecedor, não da operação.
- **As versões continuam separadas.** A aplicação vive em uma camada própria, de modo que atualizá-la sem tocar no sistema é rotina, e o contrário também.
- **A atualização entra inteira ou não entra.** A nova versão é colocada ao lado da que está rodando e ativada por uma única linha no carregador de boot. Não existe estado pela metade, e reverter é devolver essa mesma linha.
- **Repositórios externos não são necessários.** Em serviço a máquina não instala pacotes: tudo o que ela tem chegou na imagem.

Do que o sistema é composto

Parte	Função
Camada base	Linux com systemd, o daemon de gestão, a interface web, os módulos do kernel
Kernel e initrd	um kernel compilado para a máquina de destino e o script de boot inicial
Configurador	o daemon de gestão: uma API HTTP que configura a máquina através do systemd
Interface web	visão geral da máquina, interfaces, sistema, atualizações, energia
Camada da aplicação	o servidor de mídia (Mcaster, por exemplo) como camada própria sobre a base
Camada `postgres`	componente compartilhado para os produtos que precisam de um banco de dados

Duas versões

O sistema operacional tem seu próprio número de versão e o firmware do servidor tem outro. O firmware é o InfraMedia de uma versão mais a aplicação de outra; ambas ficam registradas dentro do arquivo de boot e aparecem na interface da máquina. O número do sistema operacional não se move porque a aplicação foi atualizada.

Mapa da documentação

Esta documentação vai ajudar você a:

- Entender como o sistema é construído
- Instalar o firmware em um servidor a partir de um pendrive
- Atualizar o firmware e voltar para a versão anterior

2.1 Arquitetura técnica

O InfraMedia é construído como imagem e não é modificado na máquina. A raiz do sistema de arquivos é composta de camadas, cada camada uma imagem squashfs empacotada na compilação. Tudo o que a máquina escreve fica em partições separadas.

2.1.1 Camadas do firmware

Camada	Conteúdo
Base	Linux com systemd, o daemon de gestão, a interface web, os módulos do kernel
`postgres`	o servidor de banco de dados para os produtos que precisam dele
Aplicação	o servidor de mídia: seus executáveis e o que precisarem além da camada base

A camada da aplicação é um incremento sobre a base: só entra nela a diferença em relação à camada base. As bibliotecas que já existem embaixo não sobem de novo.

2.1.2 Partições do disco

O instalador divide o disco em três partições.

- **FIRMWARE** — a partição de boot EFI: os arquivos de boot das versões instaladas, as imagens das camadas, o menu do carregador.
- **SETTINGS** — as configurações da máquina: o repositório do configurador e a sobreposição de `/etc`.
- **VAR** — dados mutáveis: logs, o diretório do banco de dados, os arquivos de trabalho da aplicação.

A raiz permanece somente leitura o tempo todo. O diretório pessoal do `root` é um tmpfs e vive até a próxima reinicialização; o que precisa sobreviver a ela fica na partição de dados mutáveis.

2.1.3 Boot

A máquina inicia pelo systemd-boot. O kernel, o initrd, a linha de comando e o **manifesto do conteúdo do firmware** estão colados em um único arquivo de boot. Isso é feito por uma única propriedade: escolher uma entrada de boot escolhe o kernel, o initrd e a lista de camadas juntos, e eles não podem sair de sincronia. Duas versões do firmware no disco são duas entradas do menu, e reverter devolve o conjunto inteiro.

O boot corre assim.

1. A UEFI entrega o controle ao carregador, que inicia o arquivo de boot da versão escolhida no menu.
2. O initrd lê o manifesto, localiza as imagens das camadas na partição de boot e **confere os sha256** contra o manifesto.
3. As camadas são montadas e compostas na raiz; por cima é montada a sobreposição de `/etc` vinda da partição de configurações.
4. O controle passa ao systemd, que sobe o daemon de gestão e a aplicação.

O initrd é um script busybox: não depende da aplicação e é compilado uma vez por arquitetura.

2.1.4 O configurador

O configurador é o daemon de gestão da máquina. Ele escuta HTTP na porta 8090 e expõe a API: visão geral da máquina, interfaces de rede, nome da máquina e servidores de hora, firmware, reinicialização e desligamento. Ele também serve a interface web — não existe porta separada para ela.

Ele age sobre o sistema chamando os programas padrão e escrevendo arquivos de configuração — `ip`, `networkctl`, `resolvectl`, `hostnamectl`, `timedatectl`, `systemctl` e os arquivos em `/etc/systemd/`.

O repositório de configurações

A única fonte da verdade sobre como a máquina está configurada é o repositório de configurações do configurador. Os arquivos do systemd são gerados a partir dele e **nunca são relidos**.

Daí decorrem duas regras práticas.

- Editar os arquivos gerados à mão não adianta: a próxima geração restaura o valor guardado.
- O valor guardado e o estado real são coisas distintas. O valor guardado vem do repositório; o estado real o configurador tira do sistema em execução pela saída legível por máquina dos utilitários do sistema, e os dois podem divergir.

A sessão de alteração

As alterações de rede são aplicadas no modo «confirme ou eu reverto». Uma alteração abre uma sessão e, enquanto ela está aberta, o repositório fica congelado por inteiro. Se a confirmação não chegar a tempo, o repositório volta à cópia guardada e as configurações às anteriores. Nunca há mais de uma sessão aberta.

O sentido do mecanismo é prático: um erro na configuração de uma interface não significa mais perder o acesso à máquina.

Modo limitado

Se o repositório não puder ser lido — o arquivo está corrompido ou a versão dele é mais nova que o próprio daemon — o configurador entra em modo limitado: as operações de leitura funcionam, as alterações são recusadas e nada é gerado. A máquina continua funcionando com o que já foi gerado.

2.1.5 O perfil de hardware

O perfil de hardware descreve as máquinas em que o firmware desta arquitetura roda: a configuração do kernel com os drivers delas e as regras que levam as interfaces físicas a nomes canônicos pelo caminho do dispositivo. Os drivers de todas as máquinas suportadas estão embutidos em um único kernel, e a mesma imagem inicia em qualquer uma delas.

Os nomes canônicos são atribuídos pelo papel da interface — `manage` para as de gestão, `streaming` para as de transmissão, com um número de ordem. Só se pode configurar pela API uma interface que tenha nome canônico; uma interface física sem ele aparece, mas não é configurável.

2.1.6 A interface web

A interface web viaja na camada base e é servida pelo próprio daemon — ela precisa funcionar em uma máquina onde a aplicação não existe ou não subiu. São cinco seções: **Visão geral**, **Interfaces**, **Sistema**, **Atualizações**, **Energia**.

As telas são construídas separadas da casca: não carregam tema, roteador nem autenticação, e não sabem onde fica a API. Isso lhes dá duas entradas — a aplicação na porta do configurador e uma seção dentro do console da aplicação, que carrega as telas da própria máquina. O operador configura o chassi onde já trabalha com o servidor de mídia.

2.1.7 Duas versões

O sistema operacional e o firmware do servidor carregam números diferentes. O firmware é o InfraMedia de uma versão mais a aplicação de outra; ambas ficam registradas no manifesto do arquivo de boot. A aplicação é compilada sobre as partes publicadas do sistema operacional, em vez de recompilá-lo.

3. Administrador

3.1 Primeiros passos

3.1.1 Instalação

O firmware é instalado no servidor a partir de um pendrive de instalação. O instalador não faz perguntas nem espera nenhuma tecla: tudo o que precisa ele tira de um arquivo de texto do próprio pendrive. Uma pessoa grava a imagem, preenche o arquivo, conecta o pendrive ao servidor e liga a energia.

O pendrive de instalação

A imagem carrega o mesmo firmware que vai parar no disco da máquina, e duas partições.

- **INSTALLER** — a partição de boot com o firmware.
- **CONFIG** — uma partição de dados comum com o arquivo `install.txt`. macOS e Windows a mostram sozinhos; a partição de boot eles escondem.

A imagem não depende de nenhum segredo: ela é construída com um modelo de `install.txt`, e o arquivo é preenchido já no pendrive, em qualquer editor. As quebras de linha e a marca de ordem de bytes que Windows e macOS acrescentam o instalador descarta sozinho.

Grave a imagem no pendrive e reconecte-o: o volume `CONFIG` aparece.

O arquivo install.txt

Abra o `install.txt` no volume `CONFIG` e substitua os marcadores entre sinais de menor e maior.

Campo	Significado
`SALT`	o sal do parque: o segredo do qual se deriva a senha de `root` da máquina. Fica no pendrive e nunca chega ao disco
`EDIT_AUTH_LOGIN`	o nome da conta para o console da aplicação e as configurações do chassi
`EDIT_AUTH_PASSWORD`	a senha dela
`LICENSE_KEY`	a chave de licença da aplicação
`HOSTNAME`	o nome da máquina
`INSTALL_DISK`	o disco onde instalar. Se omitido, é tomado o maior disco que não seja o próprio pendrive

Um marcador deixado no lugar é um valor vazio. Sem o sal ou sem a senha o instalador recusa **antes de tocar no disco**.

A instalação

Coloque o pendrive no servidor e inicie por ele: no menu de boot é a entrada `UEFI: <pendrive>`.

A partir daí o instalador trabalha sozinho:

1. divide o disco da máquina nas partições FIRMWARE, SETTINGS e VAR, **apagando-o por completo**;
2. copia o firmware para ele;
3. deixa no disco o arquivo de valores iniciais: a conta do console, a chave de licença e o hash da senha de `root`;
4. imprime no console o disco escolhido e o endereço de hardware da máquina;
5. desliga a máquina.

O pendrive pode ficar no lugar: no boot seguinte o script inicial vê a mesma versão no disco e inicia o disco. Um pendrive com outra versão instala essa de novo — uma reinstalação é feita do mesmo jeito.

A senha de root

A senha de `root` de cada máquina é derivada do sal do parque e do endereço de hardware que o instalador imprimiu. O sal vive apenas na mídia de instalação; para a máquina fica um hash, do qual a senha não se recupera.

Daí a propriedade que importa: a senha de uma máquina específica é reproduzível por quem tiver a mídia e o identificador da máquina — sem banco de dados e sem registro algum sobre a própria máquina.

Primeiro boot

Depois da instalação, ligue o servidor. A máquina sobe o daemon de gestão e depois a aplicação.

- O configurador e sua interface web ficam na porta **8090** da máquina.
- O console da aplicação fica na porta habitual dele.

As credenciais vêm do arquivo que o instalador deixou. A senha vira hash quando o repositório de configurações é preenchido pela primeira vez, e editar o arquivo em uma máquina já instalada não muda nada.

> **Sem credenciais não há como entrar**
>
> Se os campos da conta no `install.txt` ficaram vazios, o daemon recusa **qualquer** senha, não apenas uma errada. Não há credenciais no repositório nem de onde elas possam vir: a máquina terá de ser instalada de novo.

3.2 Manutenção

3.2.1 Atualização do firmware

Uma máquina instalada é atualizada sem pendrive. O pacote de atualização é instalado **ao lado** da versão em execução, não por cima: as configurações, os dados e as senhas sobrevivem à atualização, e a versão anterior fica na partição de boot para a volta atrás.

A máquina não baixa nem instala nada por conta própria. Instalar uma atualização é sempre um pedido explícito: pela tela da interface, pela API ou à mão no console.

O pacote de atualização

O pacote é um arquivo compactado com o que vai para a partição de boot: o arquivo de boot da nova versão, as imagens das camadas dela, a entrada do menu do carregador, a descrição do produto e um manifesto de somas de verificação. Ao lado do pacote é publicada a soma de verificação dele próprio.

O canal de atualizações

A máquina fica sabendo das novas versões pelo índice de um canal de atualizações cujo endereço está **compilado na própria imagem**. Ela lê o índice e lista as versões do seu produto e da sua arquitetura; as que já estão na partição aparecem marcadas.

Uma imagem sem canal nem sabe que ele existe, e só pode ser atualizada por endereço ou por arquivo.

A instalação

PELA INTERFACE WEB

A seção **Atualizações** começa nomeando duas versões: a que está em execução e a que vai iniciar da próxima vez. Ao lado fica o botão de reinicialização — quando a versão seguinte é diferente da que está em execução, ele aparece destacado, e a tela diz o que exatamente vai mudar na reinicialização.

O índice do canal é lido ao abrir a interface: o item do menu leva uma marca quando o canal tem uma versão que não está na partição. O botão **Verificar atualizações** relê o índice.

PELA API

A resposta volta na hora e a instalação corre em segundo plano.

Instalar um pacote a partir de um arquivo:

```
curl -u admin:senha -H 'content-type: application/x-tar' \
    --data-binary @mcaster-<versão>-amd64.update.tar http://<máquina>:8090/system/firmware
```

Instalar um pacote a partir de um endereço:

```
curl -u admin:senha -H 'content-type: application/json' \
    -d '{"url":"https://…/mcaster-<versão>-amd64.update.tar"}' \
    http://<máquina>:8090/system/firmware/install
```

Ver o andamento da operação, as versões instaladas e a que inicia em seguida:

```
curl -u admin:senha http://<máquina>:8090/system/firmware
```

Listar as versões do canal de atualizações:

```
curl -u admin:senha http://<máquina>:8090/system/firmware/available
```

Reiniciar na versão nova:

```
curl -u admin:senha -X POST http://<máquina>:8090/system/reboot
```

O que acontece durante a instalação

1. O daemon descompacta o pacote na partição de dados mutáveis.
2. Confere cada arquivo contra o manifesto de somas de verificação, e o produto, a arquitetura e o perfil de hardware contra os que estão em execução.
3. Só então remonta a partição de boot para escrita e copia o que falta. As imagens de camadas compartilhadas com as versões já instaladas não são copiadas.
4. Aponta a entrada padrão do carregador para a versão nova.

Depois da instalação ficam na partição a versão em execução e a nova. As demais são retiradas junto com as imagens de camadas de que ninguém mais precisa.

A versão nova só começa a funcionar depois de reiniciar — a instalação não reinicia a máquina sozinha.

Volta à versão anterior

Reverter é escolher a versão anterior e reiniciar:

```
curl -u admin:senha -X POST http://<máquina>:8090/system/firmware/<versão>/select
curl -u admin:senha -X POST http://<máquina>:8090/system/reboot
```

Remover uma versão que não é mais necessária:

```
curl -u admin:senha -X DELETE http://<máquina>:8090/system/firmware/<versão>
```

A versão em execução e a selecionada não podem ser removidas.

Quando o daemon não está

Se o daemon não subiu ou não dá para alcançá-lo, o mesmo caminho é percorrido à mão no console como `root`. O programa `firmware-update` vive na camada base, depende apenas do interpretador de comandos e dos utilitários da camada, e faz exatamente três coisas — baixar, colocar, comutar — com as mesmas verificações que o daemon faz. Ele não remove nada.

```
firmware-update install https://…/mcaster-<versão>-amd64.update.tar   # ou um caminho para um arquivo
firmware-update list                    # versões na partição: em execução e seguinte
firmware-update select <versão>         # iniciar a versão anterior da próxima vez
systemctl reboot
```

O que ainda não existe

Não há reversão automática depois de um boot malsucedido de uma versão nova: se a máquina não subir, a entrada de boot padrão é devolvida à mão pelo menu do carregador. As máquinas das primeiras séries têm uma partição de boot dimensionada para uma única versão, e uma atualização no lugar não cabe ali — essas máquinas são reinstaladas uma vez a partir de um pendrive.