Foundry

1 Wat is Foundry?

Foundry: een interne infrastructuur tool

Wat Foundry bijzonder maakt

- Eigen workflow-systeem
- 8 integraties
- Scheduled tasks
- Automatische herbruikbaarheid met auto-registering tasks
- Dropdowns met datasets

2 Demo: node create

3 Hoe werkt Foundry?

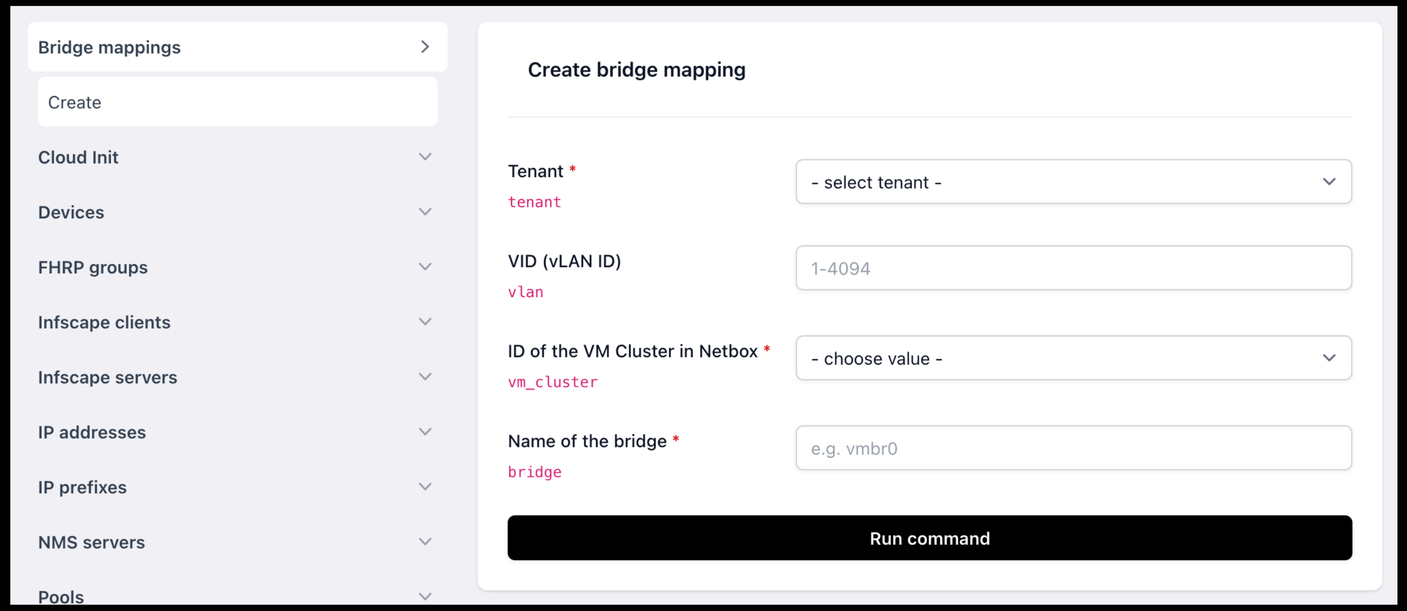
Definiëren van actions

```php
class Vm
{
    #[RequireTenant]
    #[Dataset('ip_address', [VirtualMachineDataset::class, 'cloudinitIp'])]
    public static function create(VmCreateRequest $request): CreateVm
    {
        return new CreateVm(
            data: $request->createVmPlaceholder(),
        );
    }
}
```

Reusable op 3 manieren

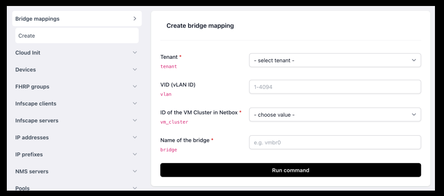
```
class Vm
{
    #[RequireTenant]
    #[Dataset('ip_address', [VirtualMachineDataset::class, 'cloudinitIp'])]
    public static function create(VmCreateRequest $request): CreateVm
    {
        return new CreateVm(
            data: $request->createVmPlaceholder(),
        );
    }
}
```

GUI



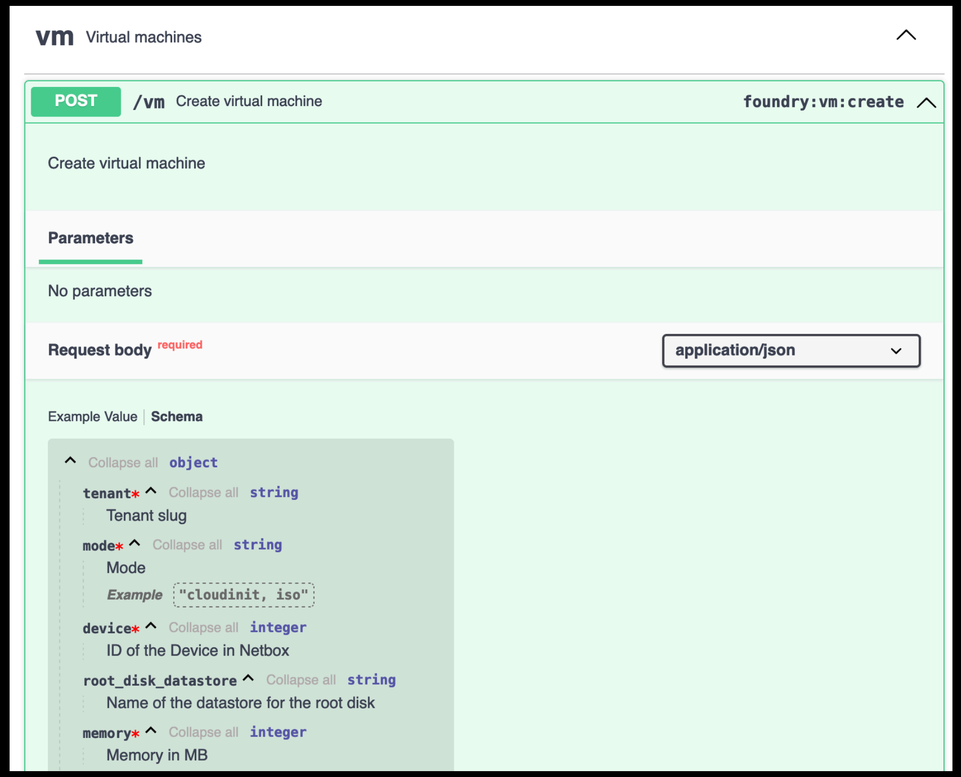
Reusable op 3 manieren

```
class Vm
{
    #[RequireTenant]
    #[Dataset('ip_address', [VirtualMachineDataset::class, 'cloudinitIp'])]
    public static function create(VmCreateRequest $request): CreateVm
    {
        return new CreateVm(
            data: $request->createVmPlaceholder(),
        );
    }
}
```



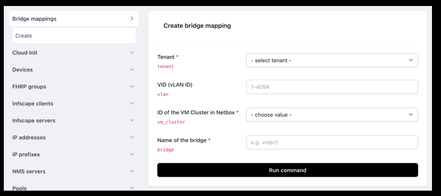
API

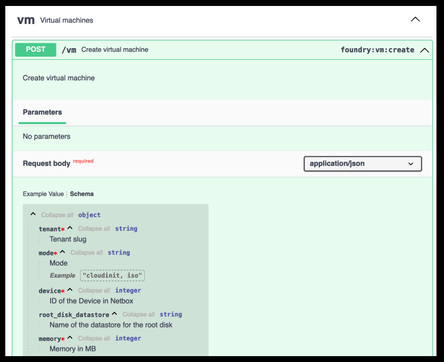
POST /api/vm



Reusable op 3 manieren

```php
class Vm
{
    #[RequireTenant]
    #[Dataset('ip_address', [VirtualMachineDataset::class, 'cloudinitIp'])]
    public static function create(VmCreateRequest $request): CreateVm
    {
        return new CreateVm(
            data: $request->createVmPlaceholder(),
        );
    }
}
```





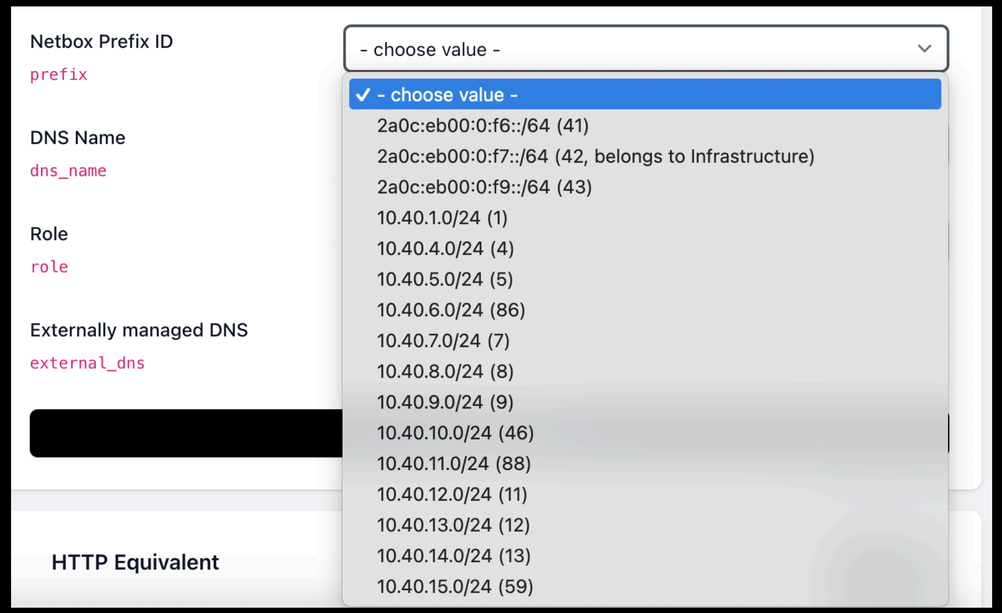
CLI

artisan foundry:bridge-mapping:create

```
foundry-staging@admin-cap:~/foundry-staging.cyberfusion.nl$ php artisan foundry:vm:create --help
Description:
  Create virtual machine

Usage:
  foundry:vm:create [options] [--] <tenant>

Arguments:
  tenant                                                Tenant slug

Options:
      --mode=MODE                                       Mode
      --device=DEVICE                                   ID of the Device in Netbox
      --root-disk-datastore=ROOT-DISK-DATASTORE         Name of the datastore for the root disk
      --memory=MEMORY                                   Memory in MB
      --hostname=HOSTNAME                               Hostname
```

Gebruikersvriendelijke GUI

```php
class Vm
{
    #[RequireTenant]
    #[Dataset('ip_address', [VirtualMachineDataset::class, 'cloudinitIp'])]
    public static function create(VmCreateRequest $request): CreateVm
    {
        return new CreateVm(
            data: $request->createVmPlaceholder(),
        );
    }
}
```



Gebruikersvriendelijke GUI

```php
class VirtualMachineDataset
{
    public static function cloudinitIp(?string $tenant, ?string $mode): Collection
    {
        if (! AbstractDataset::isTenantOk($tenant) || $mode !== 'cloudinit') {
            return collect();
        }

        return IpAddressDataset::search([
            'tenant' => $tenant,
            'status' => 'active',
            'family' => 4,
            'assigned_to_interface' => 'false',
        ])
            ->whereNull('assigned_object') // Filter IPs assigned to FHRP group
            ->map(IpAddressDataset::mapDisplay(...));
    }
```

Anatomie van een workflow

```
class CreateVm implements Workflow
{
    public function __construct(
        private readonly CreateVmPlaceholder $data,
    ) {
    }

    public static function name(): string;

    public function dependencies(): Dependencies;

    public function sequence(): Sequence;
```

Anatomie van een workflow

```
class CreateVm implements Workflow
{
    public static function name(): string
    {
        return 'Create virtual machine';
    }

    public function dependencies(): Dependencies;

    public function sequence(): Sequence;
```

Anatomie van een workflow

```
class CreateVm implements Workflow
{
    public function dependencies(): Dependencies
    {
        return new Dependencies([
            $this->tenant(...),
            $this->device(...),
            $this->vmCluster(...),
            $this->ipAddress4(...),
            $this->prefix4(...),
            $this->pool(...),
            $this->ensureDeviceHasEnoughCpus(...),
            $this->ensureDatastoreExists(...),
            $this->ensureCloudInitVendorConfigExists(...),
            $this->ensureCloudInitImageExists(...),
        ]);
    }
```

Anatomie van een workflow

```
class CreateVm implements Workflow
{

    public function dependencies(): Dependencies;

    private function pool(?Tenant $tenant, VmCluster $vmCluster): ?string
    {
        if ($this->data->pool === null) {
            return null;
        }

        $name = $tenant->poolName($this->data->pool);

        $existingPools = $vmCluster->proxmox()->pools()->get();
        $names = Arr::pluck($existingPools, 'poolid');

        if (! in_array($name, $names)) {
            throw DependencyValidationException::poolDoesNotExist('pool');
        }

        return $name;
    }

    public function sequence(): Sequence;
```

Anatomie van een workflow

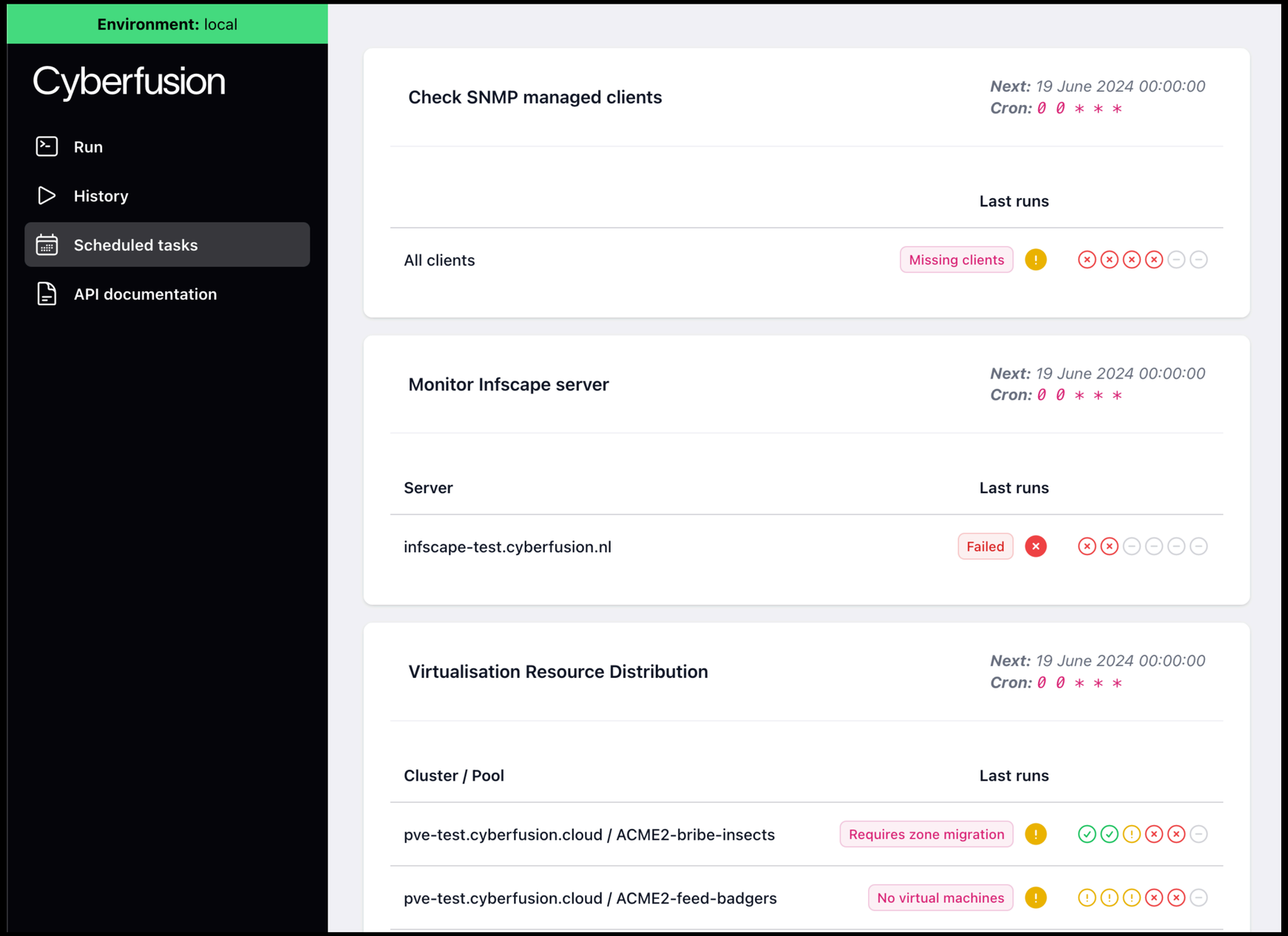
```
class CreateVm implements Workflow
{
    public function sequence(): Sequence
    {
        return new Sequence([
            self::DETERMINE_ROOT_DISK_DATASTORE => $this->determineRootDiskDatastore(...),
            self::CLOUD_INIT_DETERMINE_NETWORKING => $this->cloudInitDetermineNetworking(...),
            self::CREATE_IN_PROXMOX => $this->createVmInProxmox(...),
            self::CREATE_IN_NETBOX => $this->createVmInNetbox(...),
            self::CLOUD_INIT_CREATE_NETWORK_INTERFACE_IN_NETBOX => $this->cloudInitCreateNetworkInterfac
            self::CLOUD_INIT_ASSIGN_IP_ADDRESS_V4_IN_NETBOX => $this->cloudInitAssignVmToIpAddressVersio
            // and more
        ], $this->output(...));
    }

    private function output(array $carry)
    {
        return [
            'vm_id' => $carry[self::CREATE_IN_PROXMOX]->value()['vm_id'],
            'netbox_id' => $carry[self::CREATE_IN_NETBOX]->value()->id(),
        ];
    }
```

Anatomie van een workflow

```
class CreateVm implements Workflow
{
    public function sequence(): Sequence;

    private function cloudInitAssignVmToIpAddressVersion4(array $carry, Netbox $netbox): IpAddress
    {
        $data = $this->skipTaskUnlessCloudInit();

        /** @var VmInterface $vmInterface */
        $vmInterface = $carry[self::CLOUD_INIT_CREATE_NETWORK_INTERFACE_IN_NETBOX]->value();

        return new IpAddress(
            $netbox->ipam()->ipAddresses()->ipAddress($data->ipAddress4Id)->update([
                'assigned_object_type' => Netbox::TYPE_VM_INTERFACE,
                'assigned_object_id' => $vmInterface->id(),
            ]),
        );
    }
```

Scheduled tasks

```php
class CheckVirtualisationResourceDistribution extends ScheduledTaskJob
{
    public static function name(): string
    {
        return CheckVirtualisationResourceDistributionAction::name();
    }

    public static function column(): string
    {
        return 'Cluster / Pool';
    }

    public static function schedule(Event $schedule): void
    {
        $schedule->daily()->onOneServer();
    }

    public static function tasks(): array
    {
        return VmCluster::where('distribution_check_enabled', true)->get()
            ->flatMap(fn (VmCluster $vmCluster) => self::handleVmCluster($vmCluster))
            ->toArray();
    }
```


Environment: local
Cyberfusion
Run
History
Scheduled tasks
API documentation
Check SNMP managed clients
Next: 19 June 2024 00:00:00
Cron: 0 0 * * *
Last runs
All clients
Missing clients
Monitor Infscape server
Next: 19 June 2024 00:00:00
Cron: 0 0 * * *
Server
Last runs
infscape-test.cyberfusion.nl
Failed
Virtualisation Resource Distribution
Next: 19 June 2024 00:00:00
Cron: 0 0 * * *
Cluster / Pool
Last runs
pve-test.cyberfusion.cloud / ACME2-bribe-insects
Requires zone migration
pve-test.cyberfusion.cloud / ACME2-feed-badgers
No virtual machines

4 Waarom Laravel?

Wat gebruiken we van Laravel?

- Queues, Horizon
- Service container,
- Dependency injection
- Basics: mail, events, HTTP client
- Livewire (GUI)

Advantages

- Develop faster
- Built-in logic

5 Demo: retries

Workflows

- All actions -

Search in description

Search

Name	Status	Started at	Duration	Attempts	
Foundry presenteren *Larafest & LarAwards 2024*	Success	20 June 2024 18:00:00	10 min	1	Report
Check SNMP managed clients	Missing clients	20 June 2024 00:00:08	996 ms	1	Report
Monitor Infscape server *infscape-test.cyberfusion.nl*	Success	20 June 2024 00:00:09	90 ms	1	Report