Whalebone příručka

Vydání 3.2.1-12

20.10.2020

Contents

Whalebone je služba určená pro bezpečnostní filtraci DNS provozu. Využívá k filtraci logiku navázanou na své DNS resolvery. Resolvery mohou být buď cloudové, provozované přímo společností Whalebone, nebo lokální, provozované přímo v infrastruktuře zákazníka. Pro rozpoznávání hrozeb Whalebone využívá externí zdroje dat a vlastní algoritmy. Více informací o produktu a společnosti naleznete na oficiálních stránkách Whalebone

CHAPTER 1

Způsoby nasazení

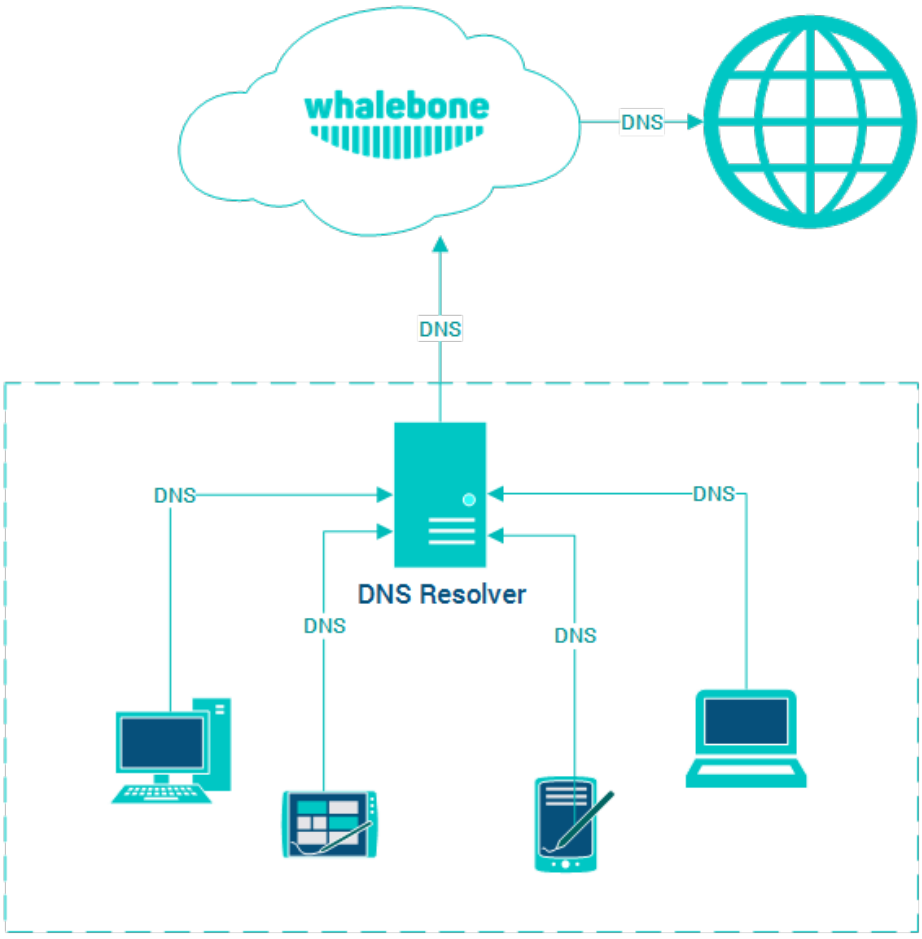
Službu Whalebone je možné nasadit a využívat v různých scénářích, které je mezi sebou možné kombinovat podle požadavků konkrétní sítě. Jedná se o kombinace využití cloudového a lokálního DNS resolveru Whalebone.

Tip: Všechny níže zmíněné způsoby nasazení lze navzájem kombinovat. Různé segmenty sítí mohou mít odlišné požadavky a jiné možnosti v provozování infrastruktury.

Tip: Pokud narazíte na problémy s nasazením Whalebone do svého prostředí a žádný z navrhovaných scénářů není vhodný, kontaktujte nás a společně navrhneme možnosti řešení vašeho konkrétního požadavku.

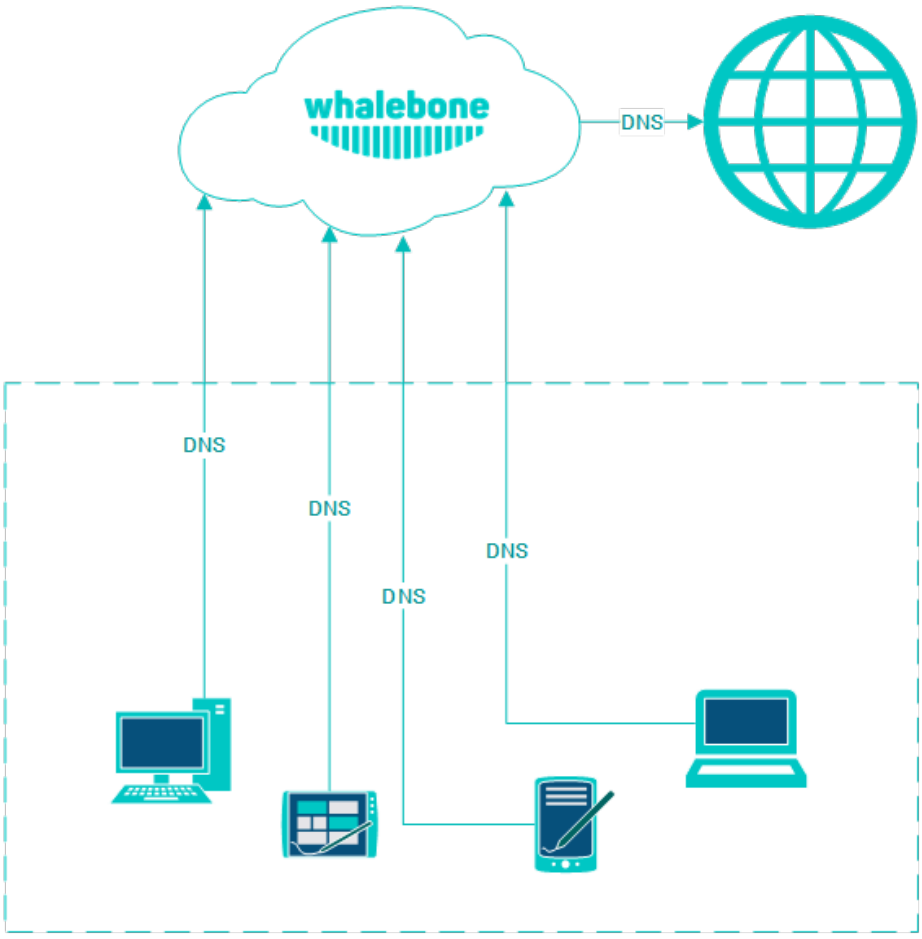
1.1 Cloud DNS

Jedná se o nejjednodušší variantu nasazení. Pro nasazení stačí úprava konfigurace aktuálních resolverů a jejich nasměrování na cloudové resolvery Whalebone. Nevýhodou tohoto nasazení je, že v případě detekce incidentu bude k dispozici pouze zdrojová IP adresa resolveru a ne původního zdrojového zařízení. Pokud je ale cílem požadavky blokovat a není důležité jednotlivá zdrojová zařízení rozlišovat, nemusí to být překážkou nasazení.



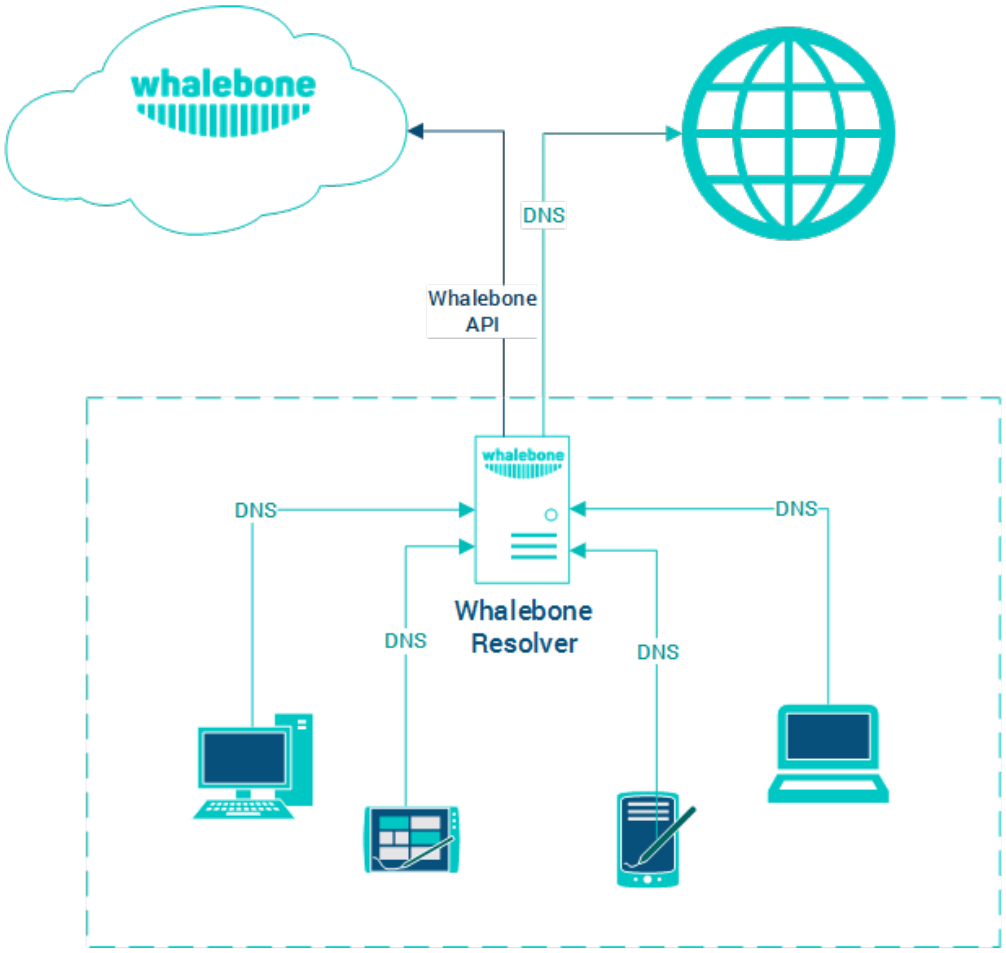
1.2 Cloud DNS (přímé spojení)

Podobný způsob jako přesměrování dotazů z vlastního resolveru na Whalebone cloud, ale dotazy jsou směrovány na cloud přímo ze zdrojových zařízení. Pro konfiguraci zařízení je ideální využít DHCP nebo jiný způsob centrální úpravy nastavení DNS překladačů. Nevýhodou tohoto způsobu nasazení je absence cache na místním resolveru, což bude mít za následek zvýšení latence překladu. Pokud nejsou jednotlivá zdrojová zařízení schovaná za překladem adres (NAT), tak budou jejich zdrojové IP viditelné v detekovaných incidentech na portálu Whalebone.



1.3 Lokální resolver

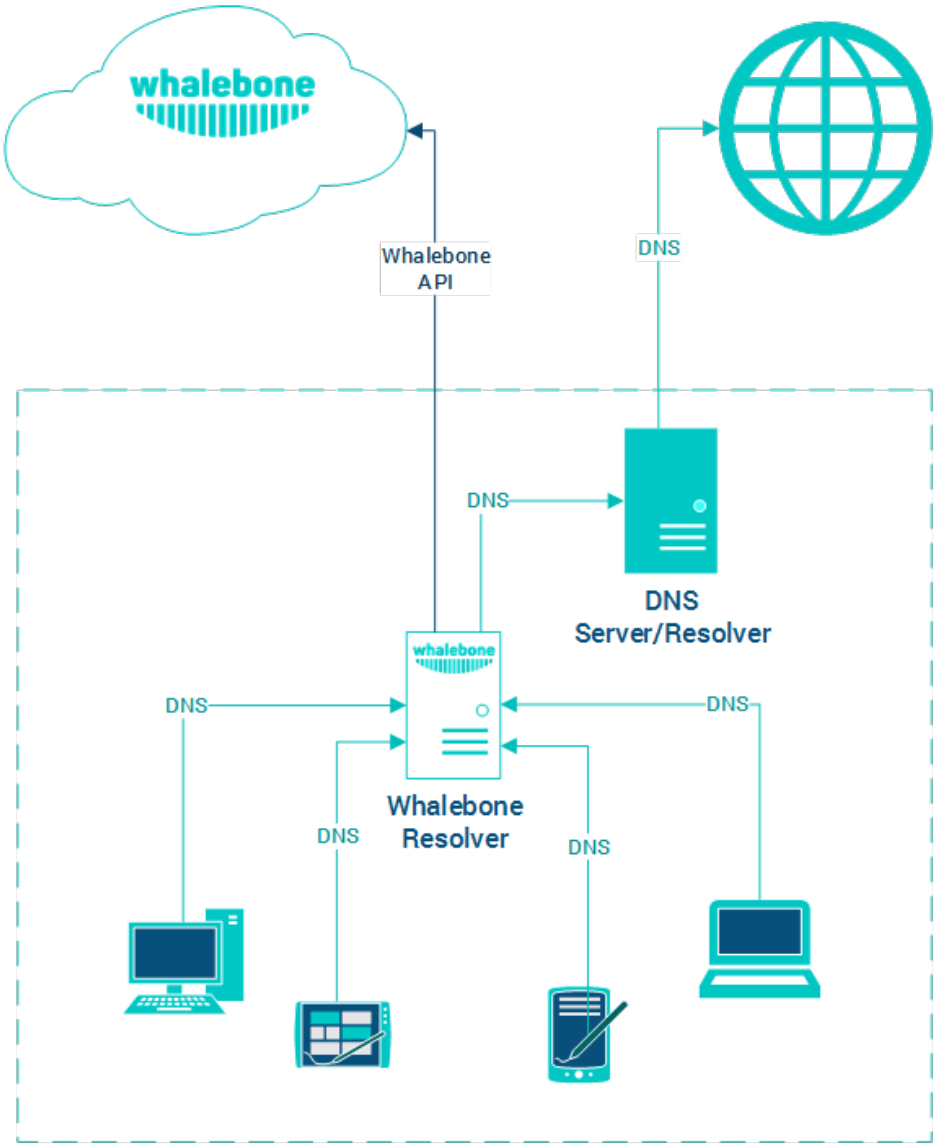
Tento způsob zapojení využívá lokálního resolveru Whalebone, který komunikuje skrze API s Whalebone cloudem. DNS překlad ale vykonává přímo a je zcela nezávislý na dostupnosti DNS překladačů Whalebone. Případný výpadek API nemá negativní dopad na dostupnost DNS překladu, ale resolver nebude schopen aktualizovat informace o hrozbách a reportovat incidenty. Hlavní výhodou tohoto způsobu nasazení je viditelnost lokálních IP adres komunikujících zařízení.



1.4 Lokální resolver (přesměrování)

Identický způsob zapojení jako v předchozím případě s tím rozdílem, že lokální resolver Whalebone nepřekládá DNS dotazy sám, ale přesměrovává dotazy na vybrané nadřazené servery. Jedná se o vhodný způsob nasazení, pokud aktuálně spravujete vlastní DNS zóny a potřebujete zajistit kontinuitu jejich překladu (např. Active Directory). Tento způsob nasazení má také nižší hardwarové nároky, cca poloviční oproti celému resolveru.

Varování: Nedoporučujeme přesměrovávat dotazy na cloudové resolvery Whalebone. Taková situace by vyústila v duplikaci detekovaných incidentů (jeden z lokálního resolveru, druhý z cloudového) aniž by tato situace přinesla vyšší úroveň zabezpečení.

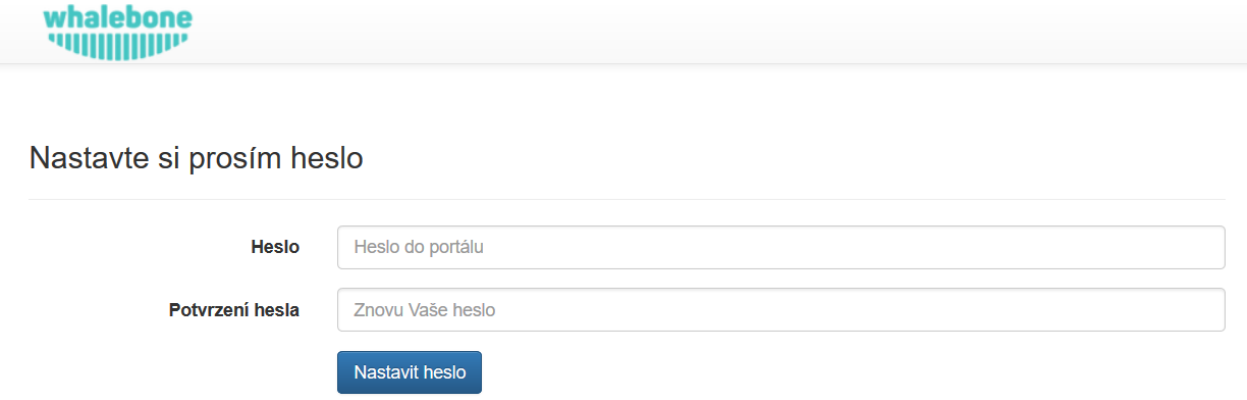

whalebone
Whalebone
API
DNS
DNS
DNS
Server/Resolver
whalebone
DNS
DNS
Whalebone
Resolver
DNS
DNS

CHAPTER 2

Začínáme

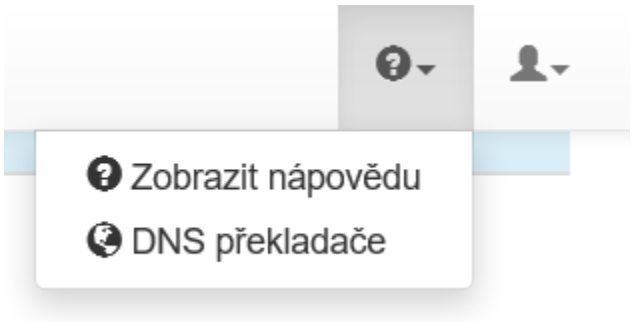
2.1 Založení účtu v portálu

Po otevření odkazu z aktivačního emailu si nastavte heslo k vašemu účtu. Nevynucujeme žádná pravidla složitosti hesla, ale doporučujeme dostatečně silné heslo pro ochranu vašeho účtu. Získáním přístupu může dojít k narušení soukromí uživatelů nebo zneužití konfigurace služby.



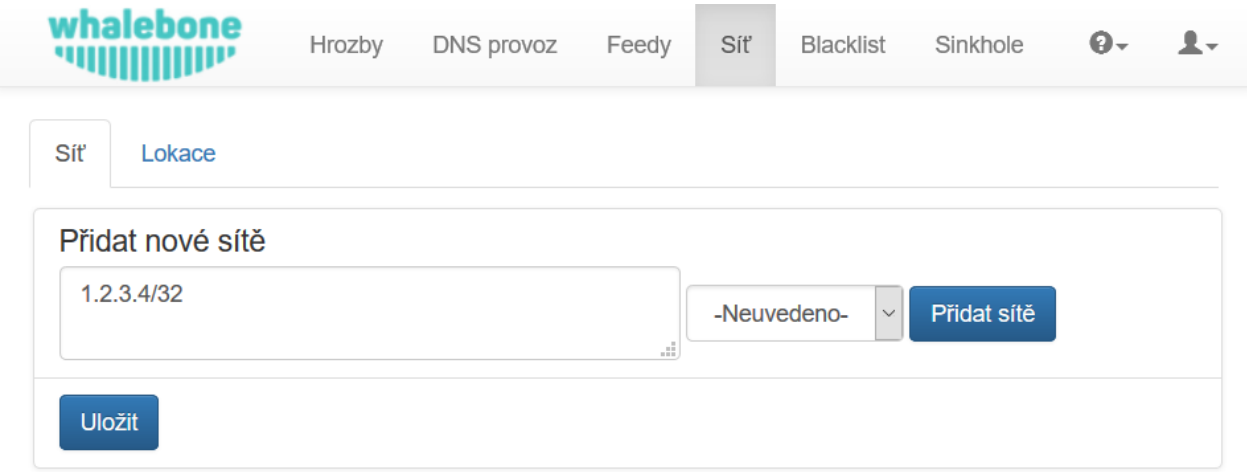
Po změně hesla budete vyzváni k prvnímu přihlášení pomocí vašeho uživatelského jména a nově zvoleného hesla.

Po prvním přihlášení se zobrazí průvodce, který vás provede hlavními možnostmi portálu. Můžete ho kdykoliv ukončit a případně znovu spustit z menu po ikonkou otazníku a volbou `Zobrazit nápovědu`.



2.2 Definice síťových rozsahů

Síťové rozsahy slouží k rozeznávání provozu jednotlivých zákazníků na cloudových resolverech a blokační stránce (sinkhole). Doporučujeme uvádět celou podsíť, ze které může přijít jak DNS provoz, tak další síťový provoz. Sítí a adres může být uvedeno více a mohou být rozčleněny do tzv. lokalit pro snazší kategorizaci DNS provozu a detekovaných událostí.



Varování: Pokud nevyplníte informace o veřejných síťových rozsazích, cloudové resolvery budou provádět pouze překlad jakéhokoliv blokování. Pokud používáte lokální resolver, jsou sítě potřebné pro správné zobrazení a customnizaci blokační stránky (sinkhole).

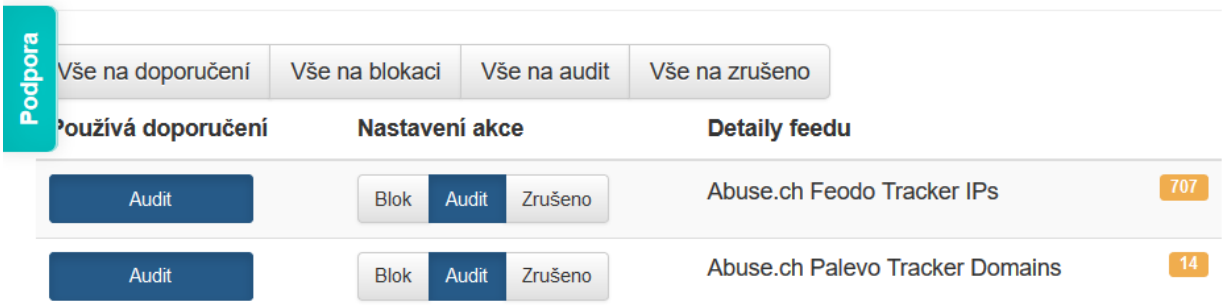
- Do pole `Přidat nové sítě` vložte jeden nebo více síťových rozsahů v notaci <adresa sítě>/<bitová maska>, např: `198.51.100.0/24`
- Stisknutím tlačítka `Přidat sítě` můžete přidávat postupné změny
- Na závěr nezapomeňte všechny změny zapsat tlačítkem `Uložit`

Tip: Při testování filtrace (např. přidáním testovací domény do vlastního blacklistu) nezapomeňte, že mnoho DNS záznamů může být aktuálně zaneseno v DNS cache kdekoliv po cestě (v browseru, operačním systému nebo resolveru). Test otevřením stránky v browseru chvíli po nasazení filtrace Whalebone může tedy selhat a doba do zapomenutí/obnovení DNS cache pro danou doménu bude závislá na velikosti TTL.

2.3 Nastavení vlastností filtrace

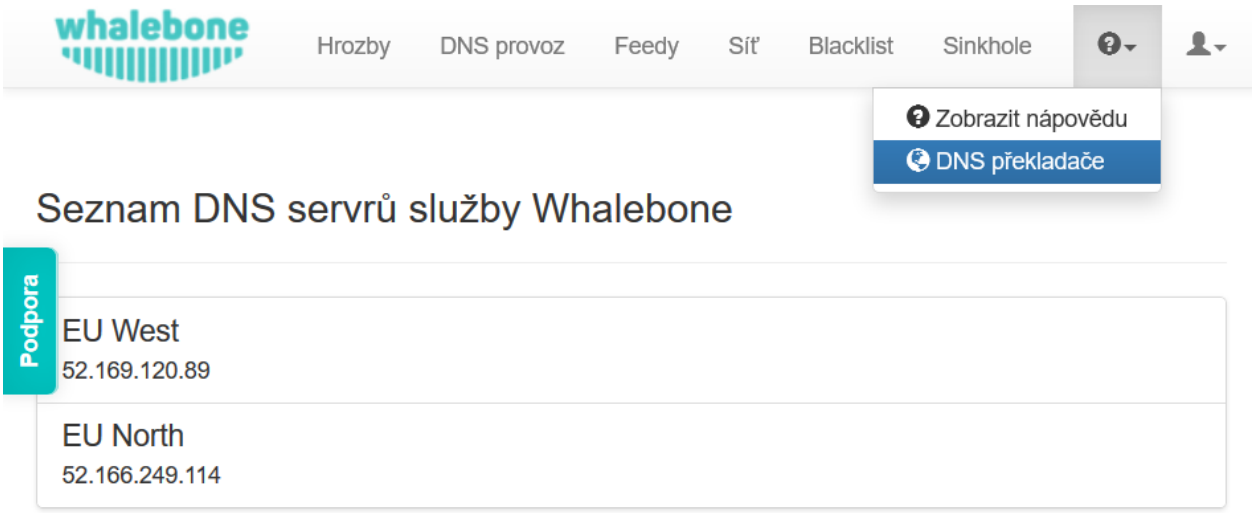
Každý zdroj informací o hrozbách (Threat Intelligence Feed) může být nastaven jiným způsobem. Pokud je stisknuté tlačítko `Používá doporučení`, řídí se nastavení doporučením provozovatele služby Whalebone. Pokud preferujete vlastní nastavení, můžete vybrat vlastní akci z možných tří voleb:

- **Blok**
- **Audit**
- **Zrušeno**



2.4 Cloudové DNS resolvery

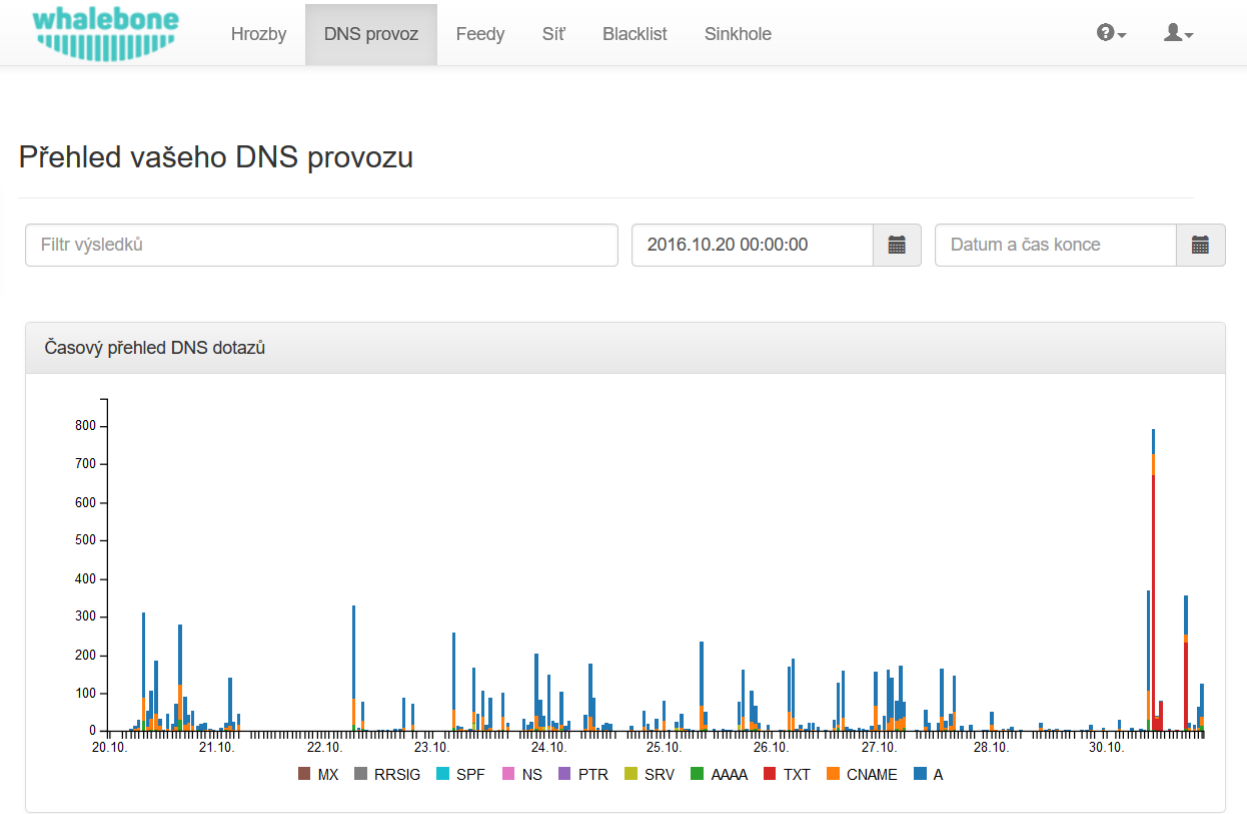
Na cloudové DNS resolvery služby Whalebone nasměrujte požadovaný provoz. Buď svých aktuálních resolverů, routerů nebo přímo jednotlivých počítačů a dalších zařízení. K dispozici jsou překladače dostupné na dvou nezávislých IP adresách: `52.169.120.89` `52.166.249.114`



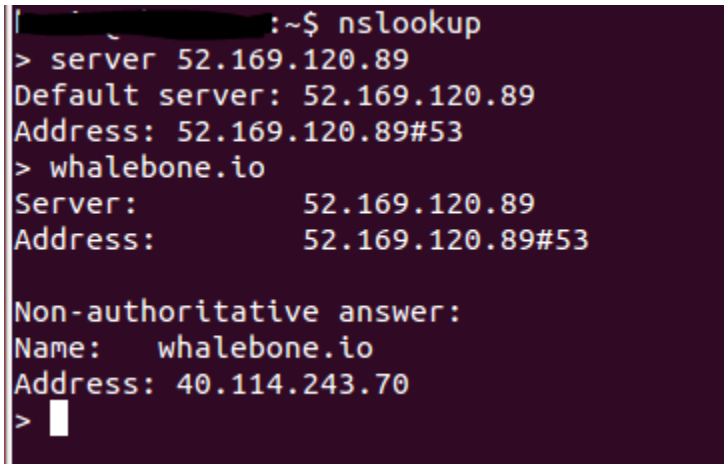
V konfiguraci vždy používejte IP adresy obou překladačů. Garance dostupnosti služby se vztahuje pouze na případy využití obou IP adres v konfiguracích, aby došlo k automatickému využití sekundárního DNS překladače při výpadku primárního.

2.5 Kontrola provozu

Jestli je provoz správně nasměrován na DNS resolvery Whalebone je možné zkontrolovat z portálu Whalebone pod položkou „DNS provoz“, kde jsou zaznamenávány jednotlivé DNS dotazy. Pokud je vše správně nakonfigurováno a funkční, bude v grafu v řádu jednotek minut viditelný DNS provoz. Pokud DNS provoz nebude na úrovni služby viditelný, překontrolujte manuálně dostupnost cloudových resolverů ze zdrojových zařízení.



Kontrolu je možné provést identicky ze strojů s OS Windows i Linux pomocí nástroje `nslookup`. Po jeho spuštění nastavte IP adresu Whalebone resolveru a zkuste přeložit doménové jméno existující domény.



CHAPTER 3

Lokální resolver

Lokální Whalebone resolver přináší oproti cloudovým resolverům zásadní výhodu ve viditelnosti konkrétních lokálních IP adres, které na něj posílají dotazy. Whalebone resolver je založen na implementaci Knot Resolveru vyvíjeného v laboratořích CZ.NIC.

3.1 Systémové požadavky

Lokální resolver předpokládá, že bude provozovován na dedikovaném stroji na čerstvě nainstalovaném a podporovaném operačním systému.

- **Podporované operační systémy** (64-bitové, serverové edice následujících distribucí):
  - Red Hat Enterprise Linux 7, 8
  - CentOS 7, 8
  - Debian 9, 10
  - Ubuntu 16.04, 18.04, 20.04
- **Podporované souborové systémy**
  - ext4
  - xfs pouze s podporou d_type (ftype=1)
- **Minimální hardwarové požadavky** (podporujeme fyzické i virtuální stroje):
  - 2 CPU jádra
  - 4 GB RAM
  - 40 GB HDD (nejméně 30 GB v oddílu /var)
- **Požadavky na síťovou komunikaci** (resolver pro svůj běh vyžaduje následující otevřené porty):
  - `TCP+UDP/53` do celého internetu pro potřeby DNS překladu

  - TCP/443 to resolverapi.whalebone.io, logger.whalebone.io, agentapi.whalebone.io, transfer.whalebone.io, portal.whalebone.io, index.docker.io, registry-1.docker.io, data.iana.org
  - Dostupnost softwarových repozitářů pro daný operační systém

Varování: Bez dostupného portu 443 na výše zmíněné destinace instalace resolveru vůbec neproběhne (instalační skript bude přerušen).

Poznámka: Kvůli odhadu sizingu pro větší podnikové a ISP sítě kontaktujte svého dodavatele. Nárůst systémových požadavků oproti standardním DNS resolverům (BIND, Unbound, apod.) se dá očekávat přibližně dvojnásobný na úrovni spotřeby RAM i zatížení CPU.

3.2 Instalace nového resolveru

V menu **Resolvery** klikněte na tlačítko **Vytvořit nový**. Zde zvolte název pro nový lokální resolver. Jedná se o čistě informativní údaj, který nemá vliv na fungování resolveru. Po vyplnění názvu klikněte na tlačítko **Vytvořit resolver** Po stisknutí tlačítka se zobrazí informativní okno se seznamem podporovaných platforem a s příkazem pro instalaci, který zkopírujte na cílový stroj a spusťte. Příkaz se postará o stažení instalačního skriptu, kterému předá jednorázový kód určený pro aktivaci lokálního resolveru (stejný příkaz nelze použít opakovaně).

Po spuštění příkazu je prováděna kontrola operačního systému a případná instalace závislostí nutných pro běh lokálního resolveru. Skript o svém průběhu interaktivně informuje a zároveň vytváří i detailní log v souboru wb_install.log v aktuálním adresáři pro případ řešení neočekávaných chyb. Úspěšný běh instalačního skriptu je zakončen oznámením Final tuning of the OS s hodnotou [OK]. Po instalaci resolveru je na pozadí ještě prováděna jeho inicializace, která může trvat až jednotky minut, než začne resolver poskytovat své služby.

Varování: Lokální resolver je nakonfigurován jako tzv. open resolver. Bude se tedy snažit vyhovět komukoliv, kdo na něj zašle svůj dotaz. To je pohodlné z pohledu zajištění dostupnosti DNS překladu všem klientům na síti, ale je nutné zajistit, aby resolver, resp. port 53 (UDP a TCP), nebyl volně dostupný z Internetu, kvůli možnému zneužití pro DoS útoky.

3.3 Bezpečnostní politiky

V menu **Konfigurace** a záložce **Bezpečnostní politiky** je možnost definovat chování filtrace DNS provozu na resolverech. Ve výchozím stavu je k dispozici **Výchozí politika**, která je automaticky přiřazována novým resolverům. V politice je možné definovat několik oblastí:

- **Filtrace nebezpečných domén**
  - Umožňuje provádět akce Audit (logování) nebo Blokaci (přesměrování na blokační stránku) přístupu na nebezpečné domény
  - Jednotlivé akce je možné úplně vypnout - např. vypnout blokaci pro testovací účely

  - Hodnota na posuvníku určuje míru jistoty, že se jedná o nebezpečnou doménu na škále 0 až 100 (0 není riziková doména, 100 je jistě nebezpečná)

Tip: Výchozí prahová hodnota blokace `80` je bezpečná i pro velké sítě s benevolentní politikou. Pro přísnější politiku ve velkých sítích doporučujeme volit blokaci v rozmezí `70-75`, velmi přísné sítě (typicky v podnikovém prostředí) si mohou dovolit blokaci až na úroveň hodnoty `60`. Audit je čistě informativní, ale příliš nízká hodnota může výrazně zvýšit počet logovaných incidentů.

- **Seznam blokovaných domén**
  - Seznamy domén, které mají být blokovány za všech okolností
  - Nemusí se jednat o rizikové domény, ale třeba o domény, které musí být blokovány na základě legislativního nařízení
  - O aktualizaci seznamů se stará společnost Whalebone
- **Výjimky**
  - Domény, které nebudou za žádných okolností blokovány
  - Výjimka se uplatňuje na danou doménu a všechny její subdomény, např.: výjimka na doménu `whalebone.io` se uplatní i na doménu `docs.whalebone.io`, ale ne naopak
- **Blokace**
  - Domény, které budou za všech okolností blokovány (vyšší prioritu mají pouze **Výjimky**)
  - Blokace se uplatňuje na danou doménu a všechny její subdomény, např.: blokace domény `malware.ninja` se uplatní i na doménu `super.malware.ninja`, ale ne naopak

Poznámka: Změny se na resolverech projeví cca do třiceti minut od uložení politik. Uložená změna konfigurace je použita pro přípravu nového balíku s informacemi o hrozbách, který si resolver z cloudu pravidelně stahuje.

3.4 Nastavení DNS překladu

V menu **Konfigurace** na záložce **DNS překlad** najdete možnosti konfigurace lokálního resolveru. Stránka umožňuje základní nastavení bez nutnosti znalosti konfigurační syntax použitého resolveru. Dále je k dispozici textové pole, které umožňuje zadat jakoukoliv konfiguraci, kterou podporuje Knot Resolver.

Dostupné možnosti konfigurace:

- **Povolit IPv6**
  - Pokud má stroj IPv6 správně nakonfigurovanou a funkční, je možné aktivovat pro resolver IPv6. V opačném případě může mít aktivace této volby negativní dopad na výkon a latenci.
- **Přesměrovat dotazy na nadřazené resolvery**
  - Tato volba umožňuje přesměrovat všechny nebo vybrané dotazy na vybrané nadřazené resolvery nebo autoritativní DNS servery (vhodné např. při přesměrování na doménové řadiče Active Directory)
  - **Zakázat DNSSEC validaci**
    - Při aktivaci této volby nebudou odpovědi z přesměrovaných dotazů validovány. Doporučujeme volbu aktivovat, pokud nadřazené servery nemají správně nakonfigurovaný DNSSEC

  - **Všechny dotazy na**
    * Možnost přesměrovat veškeré dotazy na jeden nebo více definovaných resolverů
  - **Následující domény**
    * Umožňuje zvolit konkrétní domény, které budou přesměrovány na definované resolvery
    * Je možné definovat různé resolvery pro různé domény
- **Statické záznamy**
  - Předdefinované odpovědi, které mají být vráceny na vybrané domény
  - Mohou sloužit pro speciální případy jako je monitoring, nebo velmi jednoduchá substituce vytvoření reálných záznamů na autoritativním serveru
- **Pokročilé nastavení DNS**
  - Textové pole pro plnohodnotnou konfiguraci Knot Resolveru
  - Podporuje Lua skriptování
  - Chybná konfigurace může ohrozit stabilitu, výkon a bezpečnostní funkce resolveru

---

Poznámka: Jakmile uživatel stiskne tlačítko **Uložit**, jsou změny v DNS překladu uloženy a nachystány na aplikaci na cílové resolvery. Samotné nasazení změn je ale nutné provést přímo ze stránky **Resolvery**. Je tedy možné dělat postupně více změn a aplikovat je najednou, aby se minimalizoval počet akcí zasílaných na resolver.

3.5 Správa resolverů

Na stránce **Resolvery** lze sledovat stav používaných resolverů, upravovat jejich konfiguraci, nasazovat aktualizace a instalovat nové resolvery.

3.5.1 Přehled resolverů

V hlavním přehledu resolverů jsou k dispozici dlaždice s informacemi o jednotlivých resolverech. Přehled zahrnuje informace o operačním systému a využití zdrojů jako CPU, operační paměť a diskový prostor. V přehledu je také zahrnut stav služeb běžících na resolveru (očekává se, že je „Vše v pořádku“) a stav odvozený od toho, zda resolver správně komunikuje s cloudem (pokud vše správně funguje, bude status „Aktivní“).

3.5.2 Nasazení konfigurace

Pokud jste změnili jakoukoliv konfiguraci související s logikou DNS překladu, je nutné změny na resolver manuálně nasadit. Pokud jsou k dispozici nějaké změny, které ještě nebyly na resolver nasazeny, bude v kartě viditelná červená ikonka s šipkou doprava dolů. Po kliknutí na ikonku si stránka vyžádá potvrzení, konfiguraci nasadí a zobrazí zprávu s potvrzením.

Poznámka: Pokud se při pokusu o nasazení konfigurace zobrazí chyba místo potvrzení, může jít o krátkodobý výpadek spojení mezi resolverem a cloudem, zkuste tedy akci zopakovat.

3.6 Resolver agent

3.6.1 Interakce pomocí příkazové řádky

Akce, které provádí agent, je možné volat pomocí proxy bash skriput, který se nachází v adresíři **/var/whalebone/cli**. Tento skript volá python skript, který provádí příkazy jemu předané. Tyto příkazy jsou následující:

- **sysinfo - vrací systémová data v následujícím JSON formátu**
  - Parametry: žádné
  - Výstup:


```
{
   "hostname":"hostname",
   "system":"Linux",
   "platform":"CentOS Linux 7 (Core)",
   "cpu":{
      "count":4,
      "usage":28.6
   },
   "memory":{
      "total":7.6,
      "available":3.9,
      "usage":49.2
   },
   "hdd":{
      "total":50.0,
      "free":14.4,
      "usage":71.1
   },
   "swap":{
      "total":0.0,
      "free":0.0,
      "usage":0
   },
   "resolver":{
      "answer.nxdomain":3284,
      "answer.tc":35,
      "answer.ad":849,
      "answer.100ms":3983,
      "answer.cd":6,
      "answer.1500ms":74,
      "answer.slow":215,
      "answer.rd":224337,
      "answer.1ms":104683,
      "answer.servfail":215,
      "predict.epoch":24,
      "query.dnssec":6,
      "answer.250ms":14941,
      "query.edns":35498,
      "answer.cached":86713,
      "answer.nodata":3622,
      "answer.aa":2362,
      "answer.do":6,
      "answer.edns0":35498,
      "answer.ra":224337,
      "predict.queue":0,
```


(continues on next page)

(continued from previous page)


```
      "answer.total":224337,
      "answer.10ms":35351,
      "answer.noerror":217216,
      "answer.50ms":59766,
      "answer.500ms":4642,
      "answer.1000ms":653,
      "predict.learned":80
   },
   "docker":{
      "Platform":{
         "Name":""
      },
      "Components":[
         {
            "Name":"Engine",
            "Version":"17.12.1-ce",
            "Details":{
               "ApiVersion":"1.35",
               "Arch":"amd64",
               "BuildTime":"2022-02-27T22:17:54.000000000+00:00",
               "Experimental":"false",
               "GitCommit":"88888fc6",
               "GoVersion":"go1.999.999",
               "KernelVersion":"3.22.66-693.21.1.el7.x86_64",
               "MinAPIVersion":"1.99",
               "Os":"linux"
            }
         }
      ],
      "Version":"19.32.1-ce",
      "ApiVersion":"1.98",
      "MinAPIVersion":"1.12",
      "GitCommit":"7390fc6",
      "GoVersion":"go1.9.4",
      "Os":"linux",
      "Arch":"amd64",
      "KernelVersion":"3.10.0-693.21.1.el7.x86_64",
      "BuildTime":"2018-02-27T22:17:54.000000000+00:00"
   },
   "check":{
      "resolve":"ok",
      "port":"ok"
   },
   "containers":{
      "lr-agent":"running",
      "passivedns":"running",
      "resolver":"running",
      "kresman":"running",
      "pcpy":"running",
      "logrotate":"running",
      "logstream":"running"
   },
   "images":{
      "lr-agent":"whalebone/agent:1.1.1",
      "passivedns":"whalebone/passivedns:1.1.1",
      "resolver":"whalebone/kres:1.1.1",
      "kresman":"whalebone/kresman:1.1.1",
```


(continues on next page)

(continued from previous page)


```
        "logrotate":"whalebone/logrotate:1.1.1",
        "logstream":"whalebone/logstream:1.1.1"
    },
    "error_messages":{
    },
    "interfaces":[
        {
            "name":"lo",
            "addresses":[
                "127.0.0.1",
                "::1",
                "00:00:00:00:00:00"
            ]
        },
        {
            "name":"eth0",
            "addresses":[
                "1.1.1.1",
                "::c8",
                "fe80::",
                "00:00:00:00:00:00"
            ]
        },
        {
            "name":"docker0",
            "addresses":[
                "198.1.1.1",
                "00:00:00:00:00:00"
            ]
        }
    ]
}
```


- **stop - zastaví až tři kontejnery**
  - Parametry: kontejnery, které se mají zastavit (až 3), Příklad: ./cli.sh stop resolver lr-agent kresman
  - Výstup:


```
{
        'resolver': {'status': 'success'},
        'lr-agent': {'status': 'success'},
        'kresman': {'status': 'success'}
}
```


- **remove - odtraní až 3 kontejnery**
  - Parametry: kontejnery, které se mají odstranit (až 3), Příklad: ./cli.sh remove resolver lr-agent kresman
  - Výstup:


```
{
        'resolver': {'status': 'success'},
        'lr-agent': {'status': 'success'},
        'kresman': {'status': 'success'}
}
```


- **upgrade - upgraduje až tři kontejnery, konfigrurace kontejnerů je dána docker-composem v kontejneru agenta (možné na**

  - Parametry: kontejnery, které se mají upgradovat (až 3), Příklad: ./cli.sh upgrade resolver lr-agent kresman
  - Výstup: ‘‘‘json

{‚resolver‘: {‚status‘: ‚success‘}, ‚lr-agent‘: {‚status‘: ‚success‘}, ‚kresman‘: {‚status‘: ‚success‘}} ‘‘‘

- **create - vytvoří kontejnery, konfigrurace kontejnerů je dána docker-composem v kontejneru agenta (možné najít v /etc/wl**
  - Parametry: žádné, Příklad: ./cli.sh create
  - Výstup:

```
{'resolver': {'status': 'success'}
```

- **list - zobrazí čekající příkazy a změny, který by tyto příkazy provedly na kontejnerech zmíněných v těchto příkazech, tato**
  - Parametry: žádné, Příklad: ./cli.sh list
  - Výstup:

```
------------------------------
Changes for resolver
New value for label: resolver-1.1.1
        Old value for label: resolver-1.0.0
------------------------------
```

- **run - provede čekající příkazy**
  - Parametry: žádné, Příklad: ./cli.sh run
  - Výstup:

```
{'resolver': {'status': 'success'}
```

- **delete_request - odstraní čekající příkaz**
  - Parametry: žádné, Příklad: ./cli.sh delete_request
  - Výstup:

```
Pending configuration request deleted.
```

- **updatecache - vynutí update IoC cache (používané k blokaci). Tato akce je určena pro manuální katualizaci blokovaných**
  - Parametry: žádné
  - Výstup:

```
{'status': 'success', 'message': 'Cache update successful'}
```

- **containers - lists the containers and their information which include: labels, image, name and status.**
  - Parametry: žádné
  - Výstup:

```
[
   {
      "id":"b8f4489379",
```

(continues on next page)

(continued from previous page)


```
      "image":{
         "id":"c893b4df5ca3",
         "tags":[
            "whalebone/agent:1.1.1"
         ]
      },
      "labels":{
         "lr-agent":"1.1.1"
      },
      "name":"lr-agent",
      "status":"running"
   },
   {
      "id":"e433d58f13",
      "image":{
         "id":"2c4b84a7daee",
         "tags":[
            "whalebone/passivedns:1.1.1"
         ]
      },
      "labels":{
         "passivedns":"1.1.1"
      },
      "name":"passivedns",
      "status":"running"
   },
   {
      "id":"2aeec00121",
      "image":{
         "id":"fc442e625539",
         "tags":[
            "whalebone/kres:1.1.1"
         ]
      },
      "labels":{
         "resolver":"1.1.1"
      },
      "name":"resolver",
      "status":"running"
   },
   {
      "id":"662dac2e6c",
      "image":{
         "id":"b37d0d1bd10b",
         "tags":[
            "whalebone/kresman:1.1.1"
         ]
      },
      "labels":{
         "kresman":"1.1.1"
      },
      "name":"kresman",
      "status":"running"
   },
   {
      "id":"05188ac1df",
      "image":{
```


(continues on next page)

(continued from previous page)


```
            "id":"5b50cdc924fc",
            "tags":[
               "whalebone/logrotate:1.1.1"
            ]
         },
         "labels":{
            "logrotate":"1.1.1"
         },
         "name":"logrotate",
         "status":"running"
      },
      {
         "id":"01e64dd697",
         "image":{
            "id":"fffb52c2dadd",
            "tags":[
               "whalebone/logstream:1.1.1"
            ]
         },
         "labels":{
            "logstream":"1.1.1"
         },
         "name":"logstream",
         "status":"running"
      }
]
```


Každý z představených příkazů provádí stejně pojmenovanou akci. Status a výstup této akce je zobrazován v terminálu. Akce **list** a **run** jsou určeny k řešení situací, kdy je potřeba potvrzení akcí před provedením. Akce pro zobrazení zobrazí změny, které se mají provést a kontejnery, které budou ovlivněny. Toto slouží jako náhled situace, která by se měla provést. Akce pro provedení těchto příkazu je potom provede.

Akce pro upgrade a vytvoření kontejnerů používají docker-compose, který je možné najít v kontejneru agenta, jako konfiguraci pro provádění těchto akcí. Tento soucor je připnutý v adresáři **/etc/whalebone/agent** pokud se uživatel rozhodne ho upravovat. Všechny změny musí být zaneseny i do vzoru na adrese **portal.whalebone.io**. Bez nich budou tyto lokální změny přepsány při další akci manipulující s tímto souborem.

Bash skript by měl výt volán takto: **./cli.sh action param1 param2 param3**. Action je jméno akce a jednotlivé parametry jsou parametry této akce. Pouze akce pro zastavení, odstranění a upgradování kontejnerů tyto parametry používají.

Ve výchozím nastavení agent provádí všechny změny okamžitě. Je ale možné nastavit ukládání příkazů a jejich následné ruční provádění. Díky této možnosti je možné získat větší kontrolu nad tím, které akce agent provádí. Pro zapnutí této funkcionality je nutné nastavti proměnnou prostředí **CONFIRMATION_REQUIRED** na hodnotu **true**. Pro zobrazení změn je možné použít cli akci **list**. Pro provedení uložené akce je nutné využít cli možnosti **run**. Uložený příkaz může být právě jeden, pokud přijde další, nový přepíše ten starý. Pro manuální smazání čekajícího příkazu je možné využít akci **delete_request**. Akce, které mohou být uloženy touto možností, jsou: **upgrade**, **create** a **suicide**.