DNS As A Service (DAAS) Spec

Juju Doesn't have a convenient way to interface with DNS providers. This poses an interesting problem with orchestration as it takes us 99% of the way to the finish line and falls short of setting a memorable DNS name with a given provider.

Key stakeholder

- Matt Williams (Metaswitch, UK) matt.williams@metaswitch.com
    - Matt Williams is a deployment engineer
- Canonical Cloud Strategy
    - Artur Tyloch (Teleco Program Lead)
    - Maarten Ectors (Project Lead/ Director, Cloud Strategy)

- Canonical Hyperscale
    - Sean Feole (Deployment Engineer)

User Stories

As a [PERSON], I want to [DO SOMETHING] so that [SOMETHING HAPPENS]

- As Cloud Strategist ARTUR TYLOCH, I want to EXPLAIN DNS IN TERMS OF CHARMS and demonstrate very quickly complex Juju Charm bundles like project ClearWater IMS on basically any cloud environment (also in isolated private network - without internet access) with DNS charms so that ANYONE USING JUJU CAN MANAGE THEIR DNS SIMPLY, AND AUTOMATICALLY as it helps to demonstrate Juju advanced automation and support for common deployment tasks.
- As CTO (theoretical), I want to have the capability to have DNS management of my solutions done in minutes instead of weeks by creating simple Juju relations so that

different charms get domains assigned and complex charms domain rules can be automated in a similar fashion to Route53 but work on any cloud instead of only AWS.

- As a Deployment Engineer (Matt Williams, Sean Feole), I want to MANAGE A RECORDS AND CNAMES WITH A CHARM so that MY POINTS OF ENTRY HAVE DOMAIN NAMES AND EACH UNIT OF A SCALED CLUSTER HAS A NODE DNS ENTRY.
- As a CHARM AUTHOR (Marco/Chuck/Maarten), I want a PROGRAMMABLE DNS WRAPPER WITH A CONSISTENT INTERFACE so that I CAN USE ANY SERVICE I HAVE PURCHASED DOMAIN NAMES FROM.
- As a Deployment Engineer (Matt Wiliams, Sean Feole) I want to have the ability to manage DNS in an Offline Environment, so that I'm not forced to remember/use ip addresses of my units.

Assumptions

- For Minimum Viable Product, only CNAME and A Records will be supported.
- API Wrappers will be fetched from github, as the list of providers will be dynamic and subject to rapid iteration outside of the charm.
    - This will be kept to just GoDaddy Cloud Host implementation. Route53 will be addressed pending time.
- Offline Environments will be supported with the default DNS Provider charm.
- The document will state BIND - which may be substituted for Recursive DNS Provider.
- The product itself will follow Semantic Versioning, starting with a 0.x.x series during the development cycle, a final product will introduce a 1.0.0 release. Read the semantic versioning spec here.
- Focuses on IPV4 - IPV6 and DNSSEc will be scoped and specced later.
- Using Metaswitch charms
    - https://code.launchpad.net/~projectclearwater

Risks

- DNS Server API reachability issues.
- DNS Server API changes that break compatibility.
- Undocumented technical requirements/specification.
    - We can try to mitigate, but we need feedback from Martin, Arthur, and Matt@metaswitch
- Charm Review latency can extend the overarching project goal timeline.

Design Document / Diagrams

Diagrams can be found Here

Milestones

June 6 is the Target for TADS

List milestones/checkpoints that are achievable in 2 week sprints.

- Week 1 (end on May 23)
    - **Deliverable**: Initial coding, not user facing feedback yet.
    - Scaffolding / Skeleton of Provider Charm
    - BIND / Recursive DNS Server as subset of Provider Charm
- Week 2 (end on May 30)
    - **Deliverable**: Create a basic charm relation between DNS charm and consumer (ie metaswitch) - **https://launchpad.net/~projectclearwater**
    - Implement Architecture for swapping providers
    - Feedback/Review on initial Provider Charm
    - Implement Feedback Revisions
- Week 3 (TADS: end on June 6)
    - **Deliverable**: Create a robust relation between the DNS charm and Metaswitch.
    - Feedback / Review on Provider Charm
    - Scaffolding / Skeleton of Helper Charm
        - Charm Helper can also be post-poned.
    - Implement 3rd party DNS Plugin Architecture

Post TADS

- Week 4 (TBD, pending TADS feedback)
    - **Deliverable**: Functioning GoDaddy API interface
    - Implement GoDaddy API Wrapper
    - Feedback / Review on Provider & Helper Charm
    - Finalize Bind/Recursive DNS Server
- Week 5 (TBD, pending TADS feedback)
    - **Deliverable**: Functioning: AWS Route53 API interface
    - Cycle on AWS Rt53 Provider
    - Final Deliverable with approval from Maarten Ectors and subscribed stakeholders.
- * The Time estimate is with excess time allotted for unforeseen technical difficulties. Math applied is typical work day + n hours for research / cycling on issues. (8 + n) * 2 - with time for feedback latency.

Test Plan

- Each approved model will be unit tested from day 0.
- Any incoming modules must implement tests for acceptance to parent repository
    - With repository configuration, untested modules may be deployed via 3rd party repository. This is not the default configuration.

- Amulet bundle tests will be written for key tracked deployments
- Charm hook code will be unit tested, and put under CanonicalCI

User Acceptance

- 3 of 3 test pilots sign off on the HEAD REVISION
- Deployment tests will be run, and expected behavior will be the result
- Validation will be announced via E-Mail to stakeholders

Documentation

The primary documentation for deployment of the charm (both provider and helper sub) will be provided in the README, with off-site reference to development documentation in a CONTRIBUTING.md and any additional resources determined necessary during the dev cycle.

Wrapper Documentation, Interfaces, and tech specs will be provided, and linked into aforementioned README documents. Wrappers are a layer below the charm, and should be reserved for technical people - not for those who simply wish to deploy the charm. They define an interface for integration with the charm hook code, and will rely on the wrapper author to choose a language that befits their implementation. Additional developer documentation will be provided in Markdown Format warehoused in a Gollum Wiki for public contribution of additional providers, subject to peer review prior to inclusion.

And of course I will be blogging about the experience we have introduced with DNS As A Service - both a solution and a path moving forward with community outreach.

Peer Review

- <ppetraki> +1
- <sfeole> +1 for local dns as a service
- <cmagina> +1
- <mbruzek> +1
- Maarten Ectors sign-off via Google Hangouts on May 19
- Artur Tyloch sign-off via Google Hangouts on May 19.