Release Plan

Product Name: Cloud-Native Networking
Team Member Name: Barbara Moretto (PO), Anthony Campos (SM), Gavin Moy, Katelyn Stone, Kevin Velasquez, Sabrina Tsui
Initial Date: 1/22/2019
Revision Number: 2
Revision Date: 1/24/2019

High Level Goals:
Within our project, we have two sub-projects that are similar in some parts, but vary in their high level goals. The project(s) goals are: (1) Design and implement BGP & BGP-CRD plugins for the CN-Infra platform, as well as define dependencies, builds, and deployments to Kubernetes, and (2) design and implement a web application for CN-Infra plugin integrations and generate Golang code accompanied with a visualization of internal dependencies.

User Stories (1): CONTIV/BGP-VPP

User Stories	***Story Points (1, 3, 5, 8)***
Sprint 1	
Understand Border Gateway Protocol (BGP) and how its use benefits CN-Infra.	8
Setup development environment with Goland, Vagrant, and hypervisors.	5
Understand Golang at an intermediate level, specifically with networking concepts implementations.	5
Sprint 2	
As a team, we want to have our repository nicely structured and establish good workflow and coding standards	1
As a team, we want to have a basic environment setup to run the configurations we need, so that we can begin development and testing	5
As a team, we want to become better familiar with GoBGP, so that we can incorporate into	3

our project as a library in development	
As a team, we want to implement standard operations using GoBGP, so that our pods will be able to communicate to one another on the network	8
Sprint 3	
As a team, define Makefiles, GoDep files, Docker build files, K8s deployment yamls, etc.	8
As a team, build a Go executable on top of the CN-Infra platform to get updates from ETCD about pods and VPP interfaces	8
As a team, send all updates to peers using a Go executable, and ensure all code is thoroughly tested	8
Sprint 4	
As a team, get BGP-CRD to run on it's own pod to test interaction with the network and build a Go executable for it	8
As a team, allow for BGP-CRD to get configuration from K8s, such as peers, BGP-router ID, etc	8
As a team, configure BGP-CRD with ETCD with persist config	8
As a team, get BGP status via REST and define config model	8

User Stories(2): LIGATO/OSSEUS

User Stories	***Story Points (1, 3, 5, 8)***
Sprint 1	
Understand CN-Infra to get a better grasp of the plugins from which the web application will draw from.	8
Understand React/Goland at an intermediate level, specifically with web application concepts in mind.	5
VS Code/ Goland installation and configuration	1
Understand how to use Docker and manage containers.	3
Sprint 2	
As a team, we want to have our repository nicely structured and establish good workflow and coding standards	1
As a team, we want to have a basic backend setup with default CN-Infra plugins and web server to eventually connect to our KV-store	5
As a team, we want to have a basic layout of how the frontend will look and operate, so we know what to prioritize first with components	3
As a team, we want to develop reusable react components that will be common throughout the project (i.e. drag-drop functionality)	3
As a team, we want to separate & deploy our microservices, so that any network outages/downtime doesn't cause the others to fail	5
Sprint 3	

As a team, configure TravisCI for continuous integration and create a testing suite for UI and code generator	3
As a team, create a frontend using the react components and aim to complete Palette & Plugin sections	5
As a team, begin working on code generation and define a model for two plugins that integrate together	8
As a team, integrate our frontend with our backend using the standard CRUD methodology	5
Sprint 4	
As a team, complete code generation with a test coverage of at least 80% across our system	8
As a team, begin working on inter/intra dependency map doing research and understanding how to use the plugins to accomplish this	5
As a team, finish the UI design, and have a seamless integration with good security practices from E2E	5
As a team, create any executables, dockerfiles, makefiles we might need, along with optimizing the web app for performance	8

Product Backlog:
All current user stories.