Quilt

Ethan J. Jackson

quilt.io
Compute/Network APIs

- Two Dominant Approaches
- RESTful APIs
- Static Data Serialization
  - JSON
  - YAML
Compute/Network APIs

- Traditional APIs are not expressive
- They lack
  - Composability
  - Portability
  - Flexibility
Quilt.js

A JavaScript Framework for Application Specification
Quilt.js

// `App` is a node.js application using Express, AngularJS, and MongoDB.
var App = require("github.com/NetSys/quilt/specs/mean/app");
var HaProxy = require("github.com/NetSys/quilt/specs/haproxy/haproxy").Haproxy;
var Mongo = require("github.com/NetSys/quilt/specs/mongo/mongo");

// Create 3 replicated instances of each service.
var mongo = new Mongo(3);
var app = new App(3, 8080, { MONGO_URI: mongo.uri("mean-example") });
var haproxy = new HaProxy(3, app.services(), 8080);

// Connect the app and database.
mongo.connect(27017, app);
app.connect(27017, mongo);
// Make the proxy accessible from the public internet on port 80.
haproxy.public();
Quilt.js

// `App` is a node.js application using Express, AngularJS, and MongoDB.
var App = require("github.com/NetSys/quilt/specs/mean/app");
var HaProxy = require("github.com/NetSys/quilt/specs/haproxy/haproxy").Haproxy;
var Mongo = require("github.com/NetSys/quilt/specs/mongo/mongo");

// Create 3 replicated instances of each service.
var mongo = new Mongo(3);
var app = new App(3, 8080, { MONGO_URI: mongo.uri("mean-example") });
var haproxy = new HaProxy(3, app.services(), 8080);

// Connect the app and database.
mongo.connect(27017, app);
app.connect(27017, mongo);
// Make the proxy accessible from the public internet on port 80.
haproxy.public();
Quilt.js

// `App` is a node.js application using Express, AngluarJS, and MongoDB.
var App = require("github.com/NetSys/quilt/specs/mean/app");
var HaProxy = require("github.com/NetSys/quilt/specs/haproxy/haproxy").Haproxy;
var Mongo = require("github.com/NetSys/quilt/specs/mongo/mongo");

// Create 3 replicated instances of each service.
var mongo = new Mongo(3);
var app = new App(3, 8080, { MONGO_URI: mongo.uri("mean-example") });
var haproxy = new HaProxy(3, app.services(), 8080);

// Connect the app and database.
mongo.connect(27017, app);
app.connect(27017, mongo);
// Make the proxy accessible from the public internet on port 80.
haproxy.public();
Quilt.js

// `App` is a node.js application using Express, AngularJS, and MongoDB.
var App = require("github.com/NetSys/quilt/specs/mean/app");
var HaProxy = require("github.com/NetSys/quilt/specs/haproxy/haproxy").HaProxy;
var Mongo = require("github.com/NetSys/quilt/specs/mongo/mongo");

// Create 3 replicated instances of each service.
var mongo = new Mongo(3);
var app = new App(3, 8080, { MONGO_URI: mongo.uri("mean-example") });
var haproxy = new HaProxy(3, app.services(), 8080);

// Connect the app and database.
mongo.connect(27017, app);
app.connect(27017, mongo);
// Make the proxy accessible from the public internet on port 80.
haproxy.public();

- Automatic Deployment
- AWS
- GCE
- Digital Ocean
OVN
OVN

• Quilt is just a policy layer *above* OVN
  • Replace logical switches/routers with a communication graph
  • Works great!
    • Auto-deploy OVN to the cloud!
  • Try it out! (quilt.io)
OVN — The Good

• Shockingly Stable

• Performant
  • Dataplane is slightly faster than docker
  • Control plane isn’t our bottleneck
OVN — The Bad

- OVSDB API is awkward from Go
- Hard to debug
  - Breaks rarely, but when it does . . .
Quilt

• quilt.io
  • Star the project!
  • Right now!
• Even Better …
  • Try it out!
• Give Feedback!
Thanks

quilt.io