

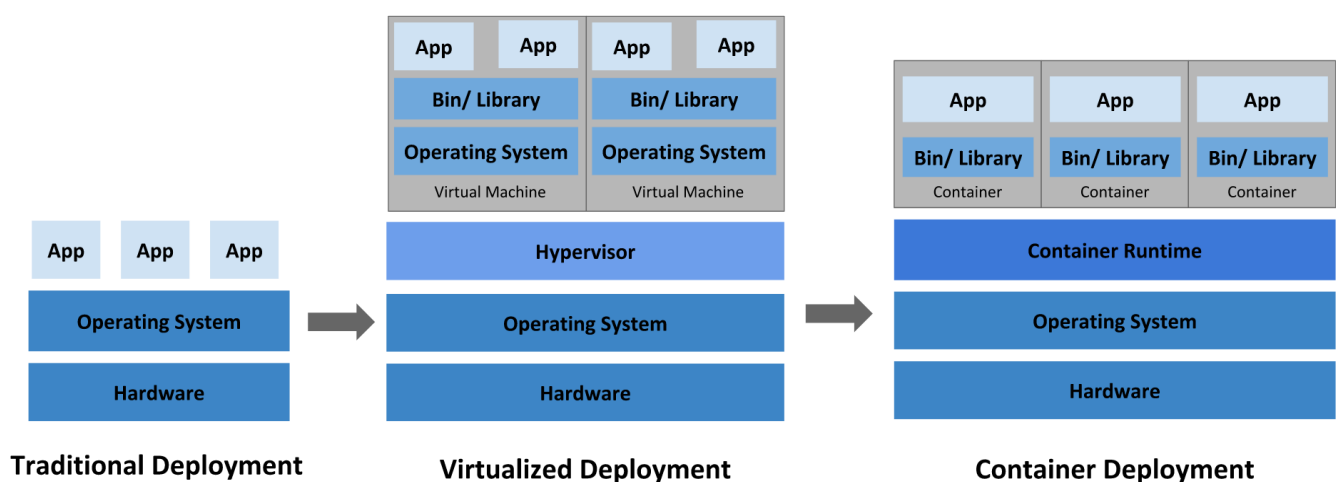
Today I want to talk about Snapt:

but wait..... lets talk about Modern service delivery and cover some lingo first.

We all have heard the words “[Digital Transformation](#)” , “[Mode 2 bimodal IT](#)”.
Businesses in 2020 are trying to use “[Digital innovation](#)” I’ve linked these terms from the first google matches I could find. In short companies are trying to use modern app delivery fabrics like Kubernetes with a set of [DevOps](#) practices.

Sounds easy, right? Well traditional application delivery applications will not support cloud ‘s micro services eccentric design.

But first, [what is Kubernetes](#)? “Kubernetes is a portable, extensible, open-source platform for managing containerized workloads and services, that facilitates both declarative configuration and automation. It has a large, rapidly growing ecosystem. Kubernetes services, support, and tools are widely available.”



As you can see above, containers allow you for a easy way to bundle and run applications.

But also adds in standards for a framework, so things like process's and tasks can be automated. Lets see what the Kubernetes website says [Kubernetes provides you with:](#)

- **Service discovery and load balancing**

Kubernetes can expose a container using the DNS name or using their own IP address. If traffic to a container is high, Kubernetes is able to load balance and distribute the network traffic so that the deployment is stable.

- **Storage orchestration**

Kubernetes allows you to automatically mount a storage system of your choice, such as local storages, public cloud providers, and more.

- **Automated rollouts and rollbacks**

You can describe the desired state for your deployed containers using Kubernetes, and it can change the actual state to the desired state at a controlled rate. For example, you can automate Kubernetes to create new containers for your deployment, remove existing containers and adopt all their resources to the new container.

- **Automatic bin packing**

You provide Kubernetes with a cluster of nodes that it can use to run containerized tasks. You tell Kubernetes how much CPU and memory (RAM) each container needs. Kubernetes can fit containers onto your nodes to make the best use of your resources.

- **Self-healing**

Kubernetes restarts containers that fail, replaces containers, kills containers that don't respond to your user-defined health check, and doesn't advertise them to clients until they are ready to serve.

- **Secret and configuration management**

Kubernetes lets you store and manage sensitive information, such as passwords, OAuth tokens, and SSH keys. You can deploy and update secrets and application configuration without rebuilding your container images, and without exposing secrets in your stack configuration

As you can see this why Kubernetes has become the default back-end of modern service delivery.

Center Stage.

Today I'm going to talk about [Snapt - Nova](#).

Snapt-Nova, Centrally Managed, Cloud Native Application Delivery

Kubernetes native ADC platform for DevOps, Developers and Infrastructure IT at companies embracing digital transformation and migrating workloads from legacy load balancers to a more modern app delivery fabric.

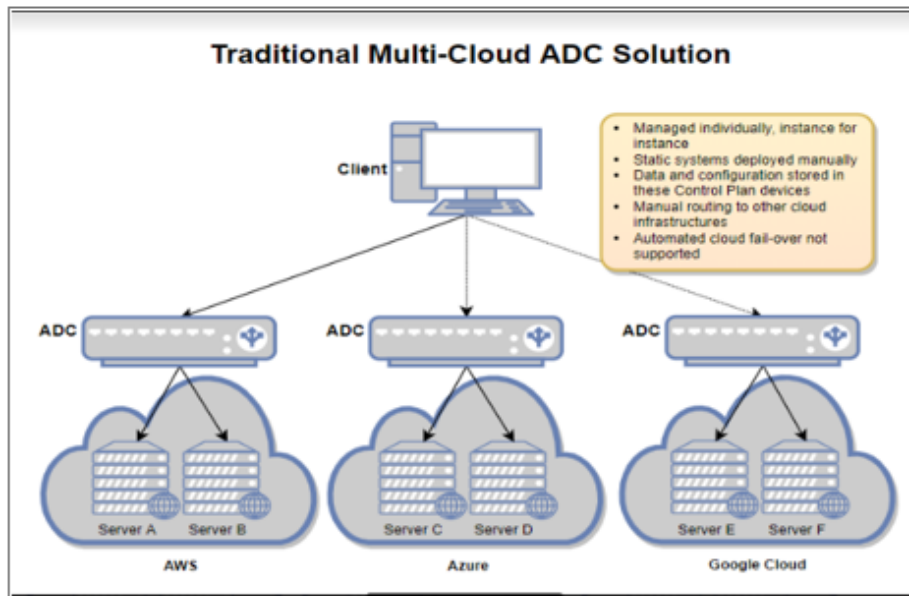
Nova features:

- Hyper scale Cloud & Kubernetes native ADC: Load Balancer, WAF, GSLB
- Easy Self-Service - Centrally Orchestrated, Configured, Deployed, Monitored
- Automatically Scale out for bursts, scale back in. No more overprovisioning
- Host on-prem or in Snapt's cloud
- Dynamic and intelligent GSLB
- Control Plane Native - Easily replicate configurations at scale
- Robust telemetry, reporting, metrics
- Deploy in any cloud and any location

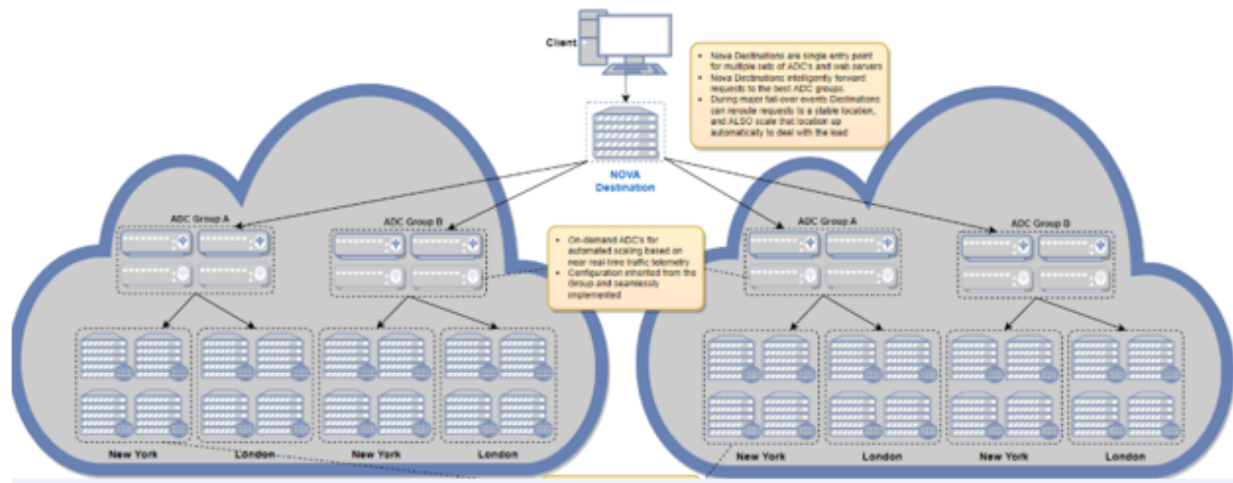
How Nova Works

- **Nova ADC Platform** - Centrally deployed anywhere, anytime. Nova deploys ADCs into any cloud, VM and container. Nova ADCs are Layer 7 load balancers, web accelerators, WAF and GSLB.
- **Nova Cloud Controller**- the Nova Cloud provides centralized deployment, configuration, management, analytics, monitoring, telemetry and more for all of your [Nova ADC](#) Nodes.
- **Nova Destinations** - destinations are powerful Nova powered intelligent DNS addresses that allow auto-scaling, traffic routing, disaster recovery, multi-cloud

routing and much more.



Nova



Use Cases/Platform

Kubernetes

Kubernetes native Load Balancer, WAF, GSLB centrally managed, and autoscaling for Kubernetes visibility, performance, and reliability.

Multi-cloud and Multi-site ADC

Snapt Nova is a centralized platform for ADC's. Deploy into any clouds or locations and manage, configurations, rote the ADCs centrally.

Cloud Native

Snapt-Nova - Cloud-Native ADC Service-discovery. Snapt-Nova features automated service discovery for Kubernetes, Consul, Docker, Rancher, more.

DigitalOcean

Snapt-Nova - the first purpose built microservices and cloud-native ADC for DigitalOcean. Droplet discovery, automated deployments and more...

Amazon AWS

Snapt for AWS EC2 is a supercharged load balancer, web accelerator and WAF much more robust than ELB. Deploy our AMI right into AWS

Web/HTTP Load Balancing

Intelligent, powerful Web Layer 7 HTTP/S Load Balancing, WAF and GSLB for your website, e-commerce store, and web apps. Free trial

Microsoft

Snapt intelligent load balancing, WAF and GSLB for Microsoft Exchange 2010, 2013 and 2016, Microsoft RDP and IIS. Snapt is a Microsoft partner.

Who is Snapt-Nova for?

- For companies embracing digital transformation that need a full featured yet right sized ADCs for different clouds, platforms, locations and budgets
- For DevOps, Developers and Infrastructure IT pro's embracing digital transformation that need a future proof and flexible on-demand application scalability for any platform and need.
- For enterprises migrating from legacy hardware load balancers to a modern app delivery fabric that is more scalable and easier to manage than legacy vendors offer.
- FOR IT looking to consolidate disparate load balancers and combine them into a

single, powerful, scalable platform for providing application delivery and visibility in clouds, Kubernetes, on-prem, vm's, anywhere and any scale

Setup and Deployment

But what's using this like? How do I sign up? What are my first setups?

- First some house keeping. I'm going to show you how easy it is to deploy a node, and even go through the first steps of the orchestration.
- This isn't a example of a best practice deployment but rather a snapshot view into the product, to allow you to see how easy and simple the product is to use and deploy.

That said, On wards!



Sign up for the free trial at <https://nova.snapt.net/>

Fill out the details and we all set to rock and roll.

You'll see a page asking about your org, type that in and click create organization.

Welcome to Nova!

As a new user, the first step is to create an organization.

<h3> Create Your Organization</h3> <p>Organization Name</p> <input type="text" value="vbrainstorm"/> <input type="button" value="Create Organisation"/>	<h3> About</h3> <p>Your organization is the ultimate holder for your servers, nodes, ADCs, users and more. You can invite other users to join your organization. Typically this would be the name of your company.</p> <p>You can change this in the future if required.</p>
---	---

Next you'll see Step one through four.

Step One

Add your first Node

A Nova Node is a system that can be controlled by Nova to run ADCs. Typically this is a server, VM, or container. If you are deploying to a cloud we advise using the Cloud Connections to speed things up.

[Create a Node](#)

[Add a Cloud](#)

Step Two

Create a Backend

Backends define upstream servers to send your traffic to, for example, a set of web servers. With Nova, backends can be dynamic as well.

[Create a Backend](#)

Step Three

Create and Deploy an ADC

You can now add a new ADC, configure it to send traffic to your Backend, and then attach it to one or more Nodes in order to run your ADC.

[Create an ADC](#)

Step Four

Learn more about the Platform

Watch our Introduction to Nova video to learn more about the power of the Nova ADC platform.

[Watch Video](#)

Let's walk through creating a node, by clicking on [Create a Node](#).

Connected Nodes Online nodes 0 Availability No Online Nodes	Activated Nodes Nodes in production 0 Percentage 0%	CPU Average processor load 0% Usage	Memory Average memory usage 0% Usage
--	--	---	--

Nodes current organisation nodes

Node	CPU	Mem	Connections	Status	Actions
Add a Node	<input type="text" value="Name"/>				Create

I'm going to add a node called, First Node, one I typed it in I hit create.

Add a Node **Create**

You will see this page, showing you Nova Installation Options.

Nova Installation Options

Nova can be installed onto any Linux compatible system. Containers, existing Linux servers, etc. Should you need help with this process please log a ticket using the Support Center. Once Nova is installed you will be able to control that server from this interface.

To proceed select an installation type below. We recommend running the docker image regardless of platform!

Your Node ID and Key are used to identify a Node, and will be required when you install. Keep these private!

```
NOVA_NODE_ID=... NOVA_KEY=...
```

Server created! Successfully added new node 'First Node'.

Manual Downloads

- Linux client
- Docker image
- Installation Guides

Sizing Guides

Unsure on how to size your virtual machine or cluster?
Check our [sizing guide](#) on the Nova Docs.

- Docker Installation recommended +
- Ubuntu Linux +
- Virtual Machine Images +

Here you can proceed with a docker install, Ubuntu Linux or grab a Virtual Machine image.

You'll also see the Manuals.

[Linux Client](#)

[Docker Image](#)

[Installation Guides](#)

[Sizing Guide](#)

I'm going to show you a example on how to deploy this with the above docker image. I have abstracted the id and key for security reasons.

```
sudo docker run --cap-add=NET_ADMIN --ulimit nofile=500000:500000 -d -  
t \  
--network=host --restart=always \  
-v /var/run/docker.sock:/var/run/docker.sock -v /etc/nova:/etc/nova \  
-e NODE_ID='putidhere' \  
-e NODE_KEY='putkeyhere' \  
-e NODE_HOST='poll.nova-adc.com' novaadc/nova-client:latest
```

sudo: docker: command not found

hmmm looks like I have to install docker. Lets do that quick.

```
sudo apt-get update && sudo apt install docker.io -y
```

```
sudo systemctl start docker && sudo systemctl enable docker
```

Unable to find image 'novaadc/nova-client:latest' locally

latest: Pulling from novaadc/nova-client

6abc03819f3e: Pull complete

05731e63f211: Pull complete

0bd67c50d6be: Pull complete

5573ecb4be96: Pull complete

049e27438cf3: Pull complete

eb98b228f2cd: Pull complete

be9c74bbddc6: Pull complete

e315c15c9b6c: Pull complete

d205ee8965c2: Pull complete

73a00ea2ce43: Pull complete

16948cade704: Pull complete

b7db21670be1: Pull complete

Digest: sha256:3c148e55f5242915f9a7d54948e6118a1481206eb4f1a0ad138d3b6a19ce1cad

Status: Downloaded newer image for novaadc/nova-client:latest


f59b400db07c59bc3bc8313f9ffdd3b22680aad49eacceb02c606115cd86066

user001@ubuntutest01:~\$

Woohoo!

Stage Right, Platform usage.

OK, now we should be set. Next navigate to ADC's, Management and Backends and lets make a backend.

 **Create a new Backend** Dr

Backend Options

Name

Choose Backend Type:

Simple
Basic backend where you specify the app server IP addresses to receive traffic. You can also set primary/backup per IP address. This is a typical load balancer configuration, and the recommended default for most users.

DNS Resolved
Basic backend with a custom resolver attached to automatically update IPs from hostnames. The difference between this and hostnames in a simple backend is the resolver which will be used to dynamically update the IPs.

Cloud API
Use a configured Cloud Connection to service discover upstream app servers by tag, image ID, etc. This allows you to use cloud technologies like autoscaling and have the backend update dynamically.

DNS Service Discovery
Use custom DNS resolvers to lookup a DNS service discovery A or SRV record. This is designed for container and autoscale environments like Docker Swarm, Consul, and Kubernetes. Recommended for advanced users only.

You'll see a result like this.

✓ Success: Backend successfully added. Please configure it as a next step.

Backend Options

Entering IP addresses. You may enter as many service addresses as you please here. Remember to add them as IP:port combinations. New IP rows will appear automatically as you fill the previous one, and there is no limit on the number of IPs.

IP:Port	Type	Weight
<input type="text" value="10.0.0.100:80"/>	<input checked="" type="radio"/> Primary <input type="radio"/> Backup	<input type="text" value="-"/> <input type="text" value="10"/> <input type="text" value="+"/>

Update

Cancel

Next we will need a ACD.

Connections Total active connections	0	Connect Rate Connections per second	0/s	Upstream Health Online percentage	0%
Requests Total requests	0	Request Rate Requests per second	0/s	Data Total data transfer	0

Below are your active ADCs. These can be deployed to active nodes in order to create live ADCs. Live statistics reset when a change is made.

Name	Nodes	Req Rate	Data	Requests	Blocked	Upstreams	Actions
------	-------	----------	------	----------	---------	-----------	---------

There are no ADCs for this organization. Please click Create below to add your first one!

Add an ADC

+ Create

Lets go ahead and create one. I just selected the first option purely as a example.

Name

http one

Choose ADC Type:

HTTP

Standard HTTP (Layer 7) load balancer with the ability to compress, cache, offload and more. Supports persistence.

SSL Termination

SSL terminated (Layer 7) HTTPS load balancer that requires certificates to offload SSL traffic and compress, cache and more.

HTTPS Passthrough

HTTPS passthrough (Layer 4) load balancer which passes HTTPS data through without termination.

Microsoft Exchange

Microsoft Exchange (2010, 2013, 2016, etc) load balancer with support for all protocols and traffic types used by Exchange.

RDP

Remote Desktop Protocol load balancer, supports Microsoft RDP, Remote Gateway, Connection Broker, and standard terminal servers.

DNS soon

DNS (TCP and UDP) load balancer providing high performance (~50,000 rps per node) DNS load balancing.

TCP

TCP (Layer 4) load balancer for custom TCP services and ports that should be tunneled through the ADC.

MySQL

MySQL database load balancer for MySQL clusters with intelligent Layer 7 health checking, for non-MySQL databases use TCP.

Submit

Cancel

Success!

✓ Success: ADC successfully added. Please configure it as a next step.

Standard

Listen & Backends

Health Checks

Performance & Security

Please select what method we will load balance traffic with and how to track the session stickiness or persistence of the connection.

Balancing Method

The method decides how to share the traffic between backends. For HTTP we recommend round robin. If you are not using persistence you may need source hashing.

Roundrobin [Recommended]

Session Persistence

This allows you to stick users to the first server they connect to, useful for authentication.

Cookie Insertion [Recommended]

As you can see you can change and configure this as needed. Next, I want to attach a node to my ADC.

You may now attach this ADC configuration to one or more Nodes or Tags. This will deploy and run the ADC on those nodes immediately.

Nodes

Select the nodes to attach this ADC to. If a node is not shown it may not be ready.

First Node

Tags

Select the tags to discover nodes for this ADC. If you are using an autoscaler select only the autoscaler you wish to use.

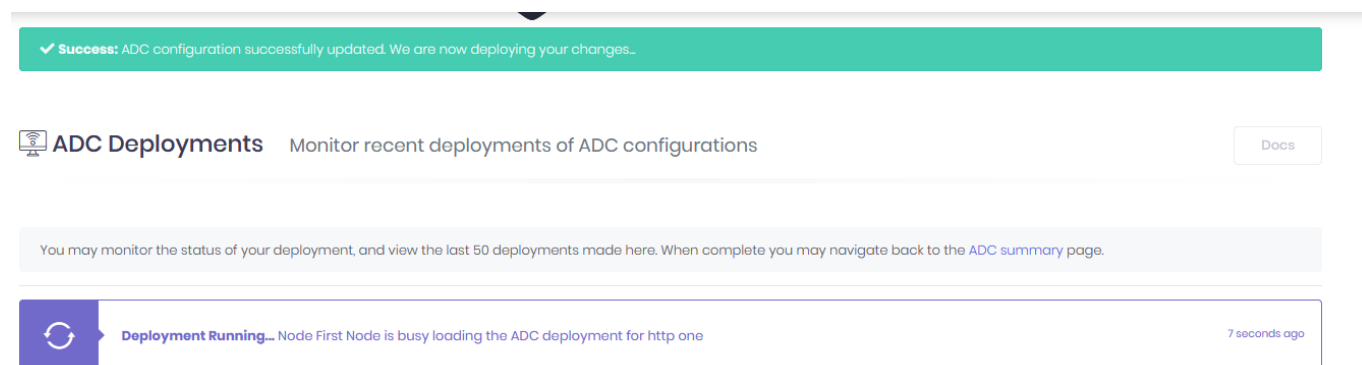
ALL NODES

Attach

Cancel

I clicked Edit. In this example, I'm just going to click my only node, and tag it with all nodes,

as it's my only tag in my example.



My Deployment is running! Now, In most cases you would be doing a full deployment , while I'm just showing you a example of how to use the product. But as you can see here is my dashboard.

I hope this example of what [Snapt - Nova](#)'s platform has to offer was interesting to you. I hope I get a chance to dive deeper into this interesting technology in the future.

Written by [Roger Lund](#)

VMware and Storage crazy man, vExpert, MN VMUG leader

