F5 NGINX

App Delivery Manager

Agility 2023

Brett Wolmarans

b.wolmarans@f5.com

Technical Product Manger

NGINX
Part of F5

ADM in a sentence:

A *GUI* *(and API)* tool for App Delivery, with separate Management and Data Planes.

- Load Balancing
- Encryption
- Proxying
- Caching
- Health Monitoring
- WAF

App Connectivity Stack

Solution Components

App Delivery Manager

Simplify the complexity of governing and managing APIs in multi-cloud environments

NGINX Management Suite

Instance Manager

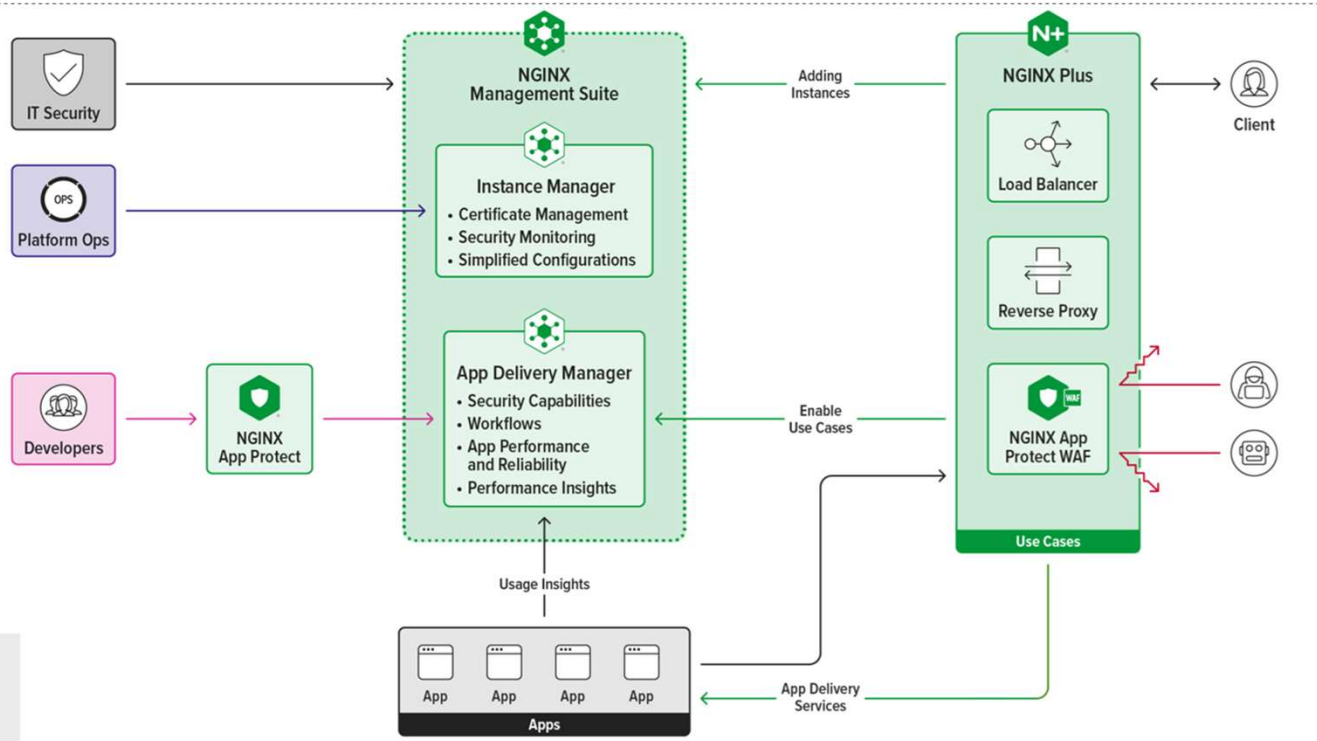
Track and control NGINX Open Source, NGINX Plus, and NGINX App Protect WAF instances

NGINX Plus

Deliver uncompromised performance and reliability with lightweight, real-time NGINX API gateways

NGINX App Protect

Ensure consistent oversight for APIs with global policies and fine-grained controls for API owners



App Delivery Evolution with NGINX

Goals Achieved

- High-level Declarative API for the ADC & API-m use cases
- API/UI supports RBAC that creates self-service model
- Platform is resilient with 3-node cluster
- Flexibility is achieved with later introduced "snippet" feature
- WAF was available for both ADC & API-m use cases per application and per API.
- Instance Group support (instances are ephemeral)
- With Forwarder support, metrics/events can be pushed to external system but at the same time Platform also stored all the data

Challenges

- Challenge with installation, updates, and upgrades
- Debugging challenge with operational issues
- Instance Group was built with BYOD (bring your own dataplane) model, that created some operational overhead
- Performance and scale limitation – UI & API slowed down after some scale
- Extensibility – though achieved through snippet, it was not a first-class feature and broken RBAC model
- Config inconsistency – We have seen some issues with DPM cache inconsistency that resulted into undesirable config on the dataplane.

INTRODUCING

NGINX Management Suite App Delivery Manager

Goal: The Application Delivery Manager module's addresses the use cases and challenges that our current customers face regarding self-service, scalability, and complex configuration management.

Objectives: Unify application delivery and security operations from a single point of control. Like Controller ADC, NGINX App Delivery Manager enables:

- Automation of application infrastructure for simple and repeatable onboarding of new teams
- Provides an app-centric user experience that enables developers to leverage the power of NGINX without deep expertise
- Seeks to serve both App Dev/DevOps' desire for speed and self-service as well as Platform Ops' need for governance, observability, and overall simplification of app delivery workflows

Differentiator: **Enable feature velocity when time-to-market matters.** With introduction of Templates, customers may extend the core ADM API and make available for use any NGINX directive to meet the needs of emerging use cases.

APP DELIVERY MANAGER CAPABILITIES

Platform Ops: Leverage a single point of control for app delivery and security

- **Centralized Lifecycle Management**. Enable segmentation of infrastructure into clearly defined domain boundaries to fully support the needs of software development.
- **System Access Controls**. Role-based access controls to **all product features and data** to ensure that teams can work independently and only use and see the services and information that they need to complete their job.
  - Includes support for OIDC compliant providers
- **Observability**. Central, aggregated view of metrics of system and app use and performance, which is vital as apps and services dynamically grow in number and complexity.
- **Decrease time to deploy modern apps**. Define a self-service model that provides the tools for developers to work independently with their preferred infrastructure while ensuring that network and security policy compliance are met.

Slide 6

RE0 Difference + Migration Vision
Execution
Roderick Escobar, 2023-03-17T16:14:36.993

RE0 0 persona?
Roderick Escobar, 2023-03-17T16:15:43.010

RE0 1 Add main problems that Controller ADC solved and ADM

Check with Eric/Rob
Roderick Escobar, 2023-03-17T16:16:29.683

Developers: Use a self-service platform to build configs, control APIs, and deploy apps and services

- **Extensibility**. Adapt and extend App Delivery Manager features to suit your organization's unique needs with **OOB and custom Templates** that enable the use of any NGINX directive to unlock use cases that accelerate app performance and deliver frictionless digital experiences for customers. Examples include:
  - Improve App Performance with Caching. Enable `proxy_cache_path` for top-level http context; `proxy_cache` to **location** blocks in nginx.conf
  - Improve HTTP Efficiency. Modify `listen` directive to include `http2`
- **Frictionless.** Develop and deploy apps and services without the need of services from external teams within the enterprise.
  - Simplify and streamline application delivery and security with the ability to reliably configure groups of NGINX Plus instances (via I**nstance Groups** and **Workload Affinity**) in an automated and repeatable fashion.
- **Automation**. Simplified set of REST APIs to support automated workflows end-to-end and integration with external tools.
- **Observability**. Near-real time insights traffic metrics per app enables insights into app and network performance to monitor delivery of digital experiences.

Architecture

How to use this section

- There are multiple architecture slides in this section, because there are multiple ways to think about it (block diagram, layers, etc.)
- Please use the one that your customer can best relate to.

NGINX Management Suite

Graphical User Interface

GitOps / Scripts / Automation Systems

API

Instance Manager

API Connectivity Manager

App Delivery Manager

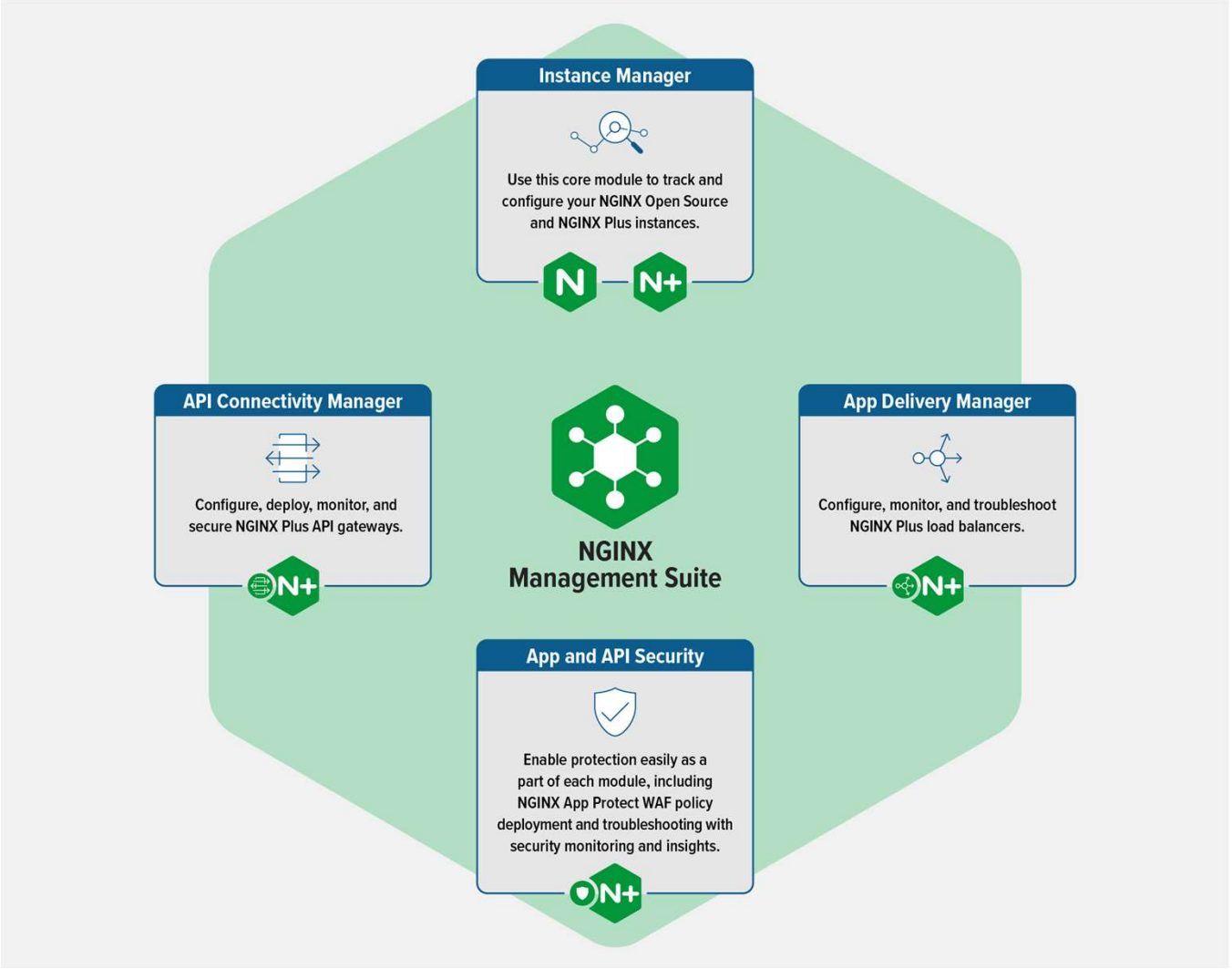
Security Monitoring

Future F5 NGINX Modules

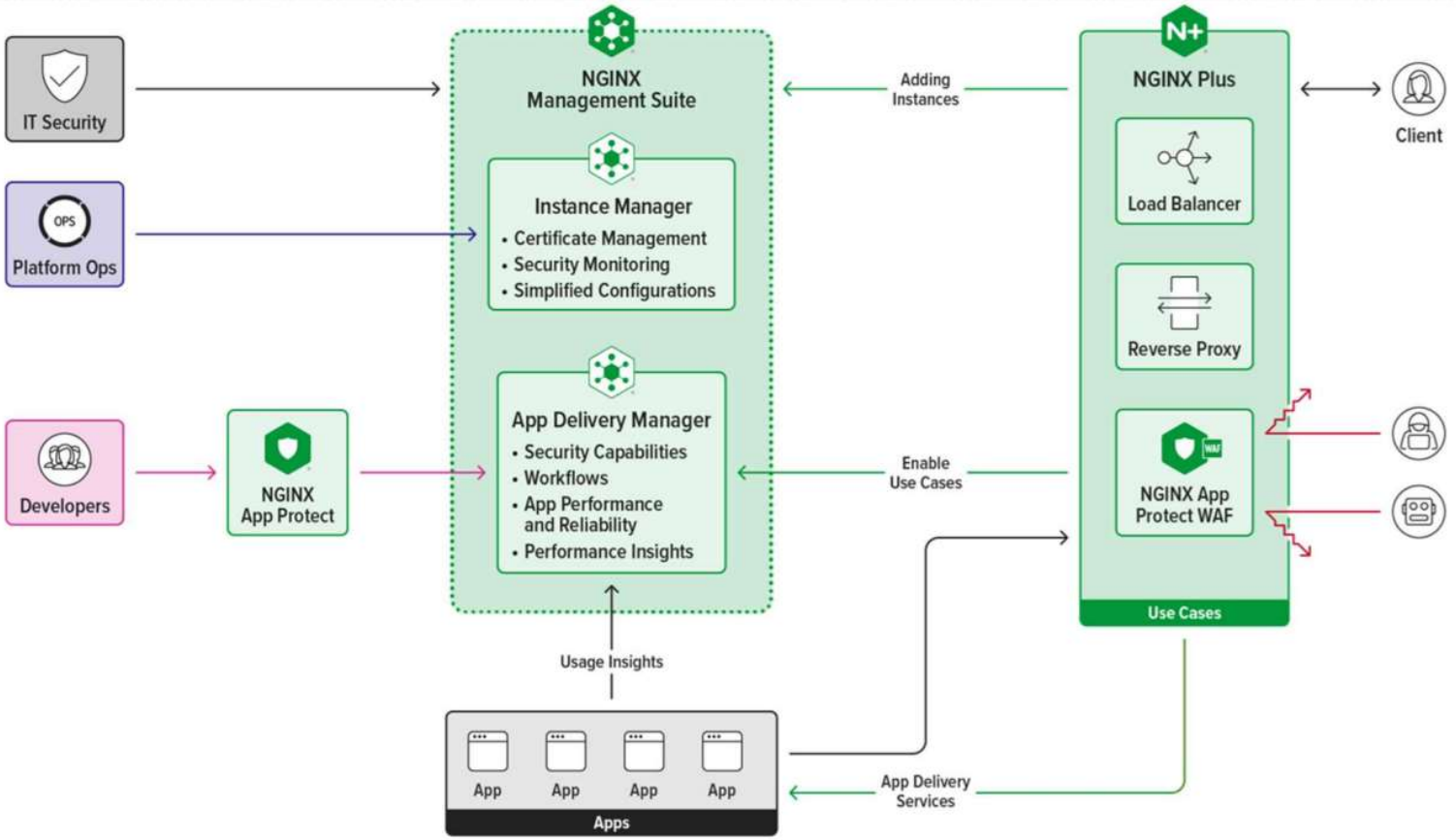
Future Partner Ecosystem Contributions

NGINX Agent

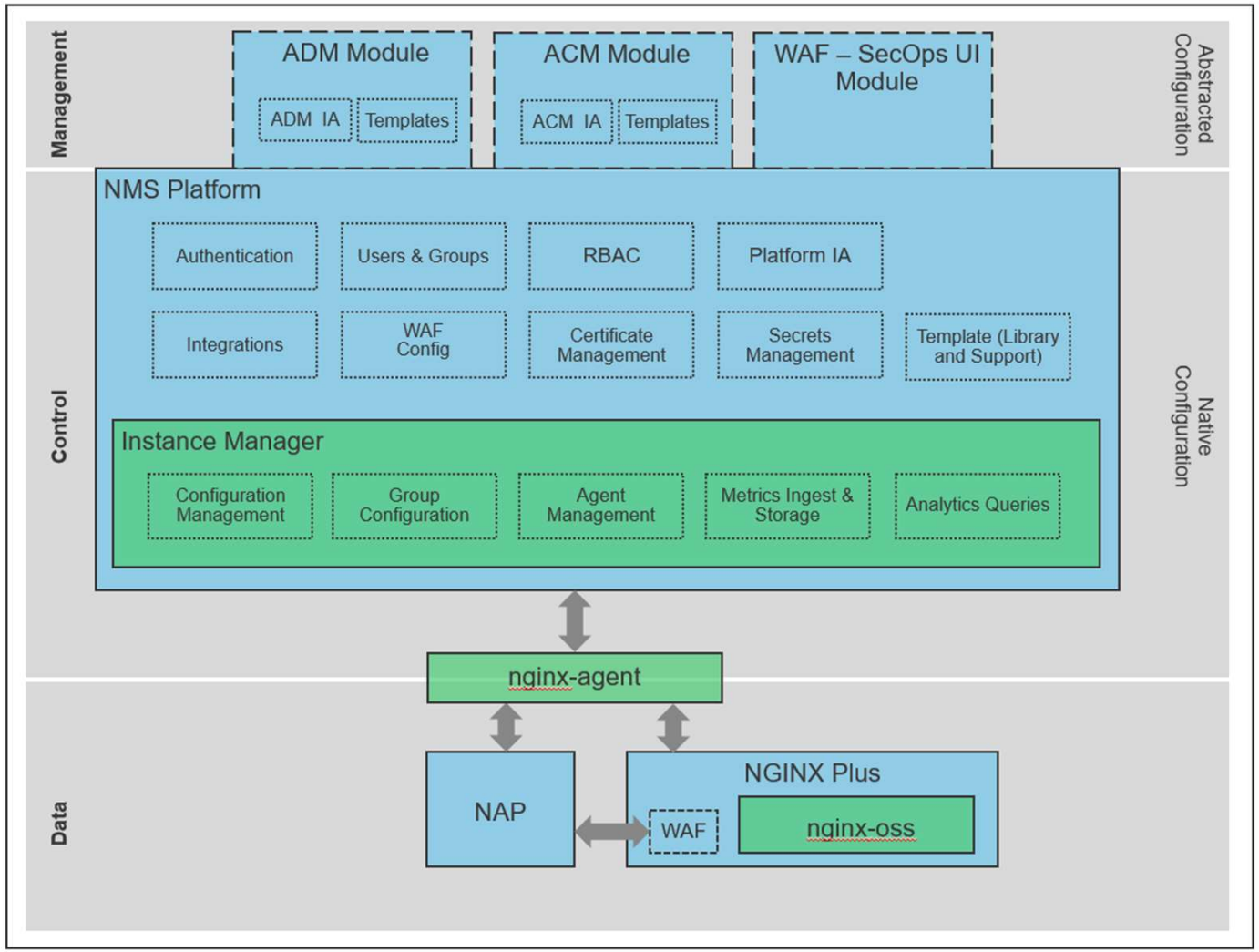
NGINX  


Instance Manager
Use this core module to track and configure your NGINX Open Source and NGINX Plus instances.
API Connectivity Manager
Configure, deploy, monitor, and secure NGINX Plus API gateways.
NGINX Management Suite
App Delivery Manager
Configure, monitor, and troubleshoot NGINX Plus load balancers.
App and API Security
Enable protection easily as a part of each module, including NGINX App Protect WAF policy deployment and troubleshooting with security monitoring and insights.

ADM High-Level Architecture



Architecture



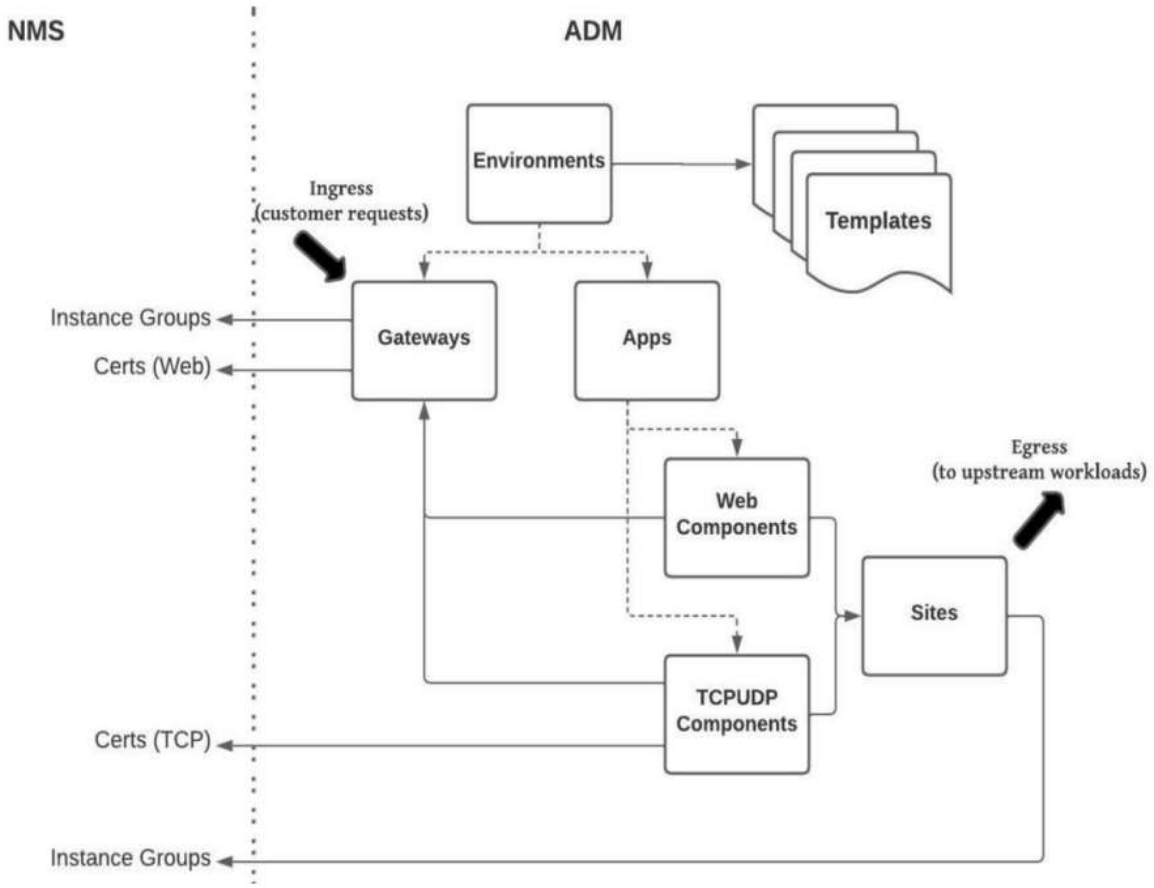
Concepts

NGINX Directives

- NGINX has 100+ directives, and permutations
- There are major sections, or blocks in an NGINX configuration
- Server Block, Location Block, and Upstread Block are the most important for delivering applications.
- ADM abstracts these into the following (next slide)

ADM Concepts

API Object	Affected NGINX Contexts	Description
Environments	-	Logical separation of objects by organizations or deployment types.
Gateways	**Server blocks**	Definition of network entry point (ingress for NGINX+) for the app(s).
Apps	-	Logical grouping of components that define an individual app.
Web components	**HTTP Location blocks**	Definition of a microservice or part of a microservice that implements a HTTP or HTTPS app.
Workload Groups	**Upstreams**	Upstreams that are used in proxy_pass from the Web Components location blocks
TCP/UDP components	**server** (stream), **upstream** (stream)	Definition of a microservice or part of a microservice that implements a TCP, TC+TLS, or UDP app.
Sites	-	Logical separation of instance groups, typically based on locality.



Templates

Templates are a way to extend ADM. They are created on the box

/etc/nms/modules/adm/

templates/

/base

/use-cases

Great power, Great Responsibility. Best left to professional services, or very advanced customers. Full control over any NGINX Directive.

Fair game for SE's/SA's/Customers to create / modify / share ☺ cannot over-write directives defined in the base templates.

Use Case Template Schema	Purpose
gateway.json	.JSON makes the GUI (fields and labels) for the user
server-gateway.tpml	.TMPL is the GO Template that takes the user input, and renders the text that will go into nginx.conf


NGINX®