

Projects

- [Project Page Templates](#)
- [Proposing a new Project](#)
- [Modules and Components](#)
- [Infrastructure](#)
- [Integration](#)
- [Training](#)
- [TF Operator Framework](#)
- [Documentation](#)
- [TF](#)

A **Project** is a collection of work undertaken to deliver a well-defined goal. The goal can be anything, it could be a feature enhancement or a big bug, support activity. Anything of significance that requires a well-defined goal can be a project. Projects transcend [modules and components](#). A project is generally associated with a timeframe, a plan, and a collection of people. A technical leader/manager (PTL) for a project **MUST** be defined.

Project PTLs should update this page with the name of the Project (Module), the PTL, and a brief project description. Once that is done, approved projects should please create a Project Page based on the [Template instructions](#). If you do not yet have an approved project, please follow the instructions for [Proposing a new Project](#) (Module). If you still have questions, please reach out to [Casey Cain](#).

If you want to enrich the TF Product with a new piece of functionality/improvement in regard to a particular module which is currently covered by an existing project, please refer to a [Blueprint approval process](#)

End to End Architecture - [Yuvaraja Mariappan](#) [Chandra Mohan](#) (This is an umbrella role)

Marketing Representative - [Adam Grochowski](#)

Project	PTL contact	Modules	Short Description
TF Core	Chandra Mohan	<ul style="list-style-type: none"> • Configuration System • Fabric Management • Control Plane • Data Plane • Kubernetes Orchestrator • Openstack Orchestrator • Deployment • Packaging • Analytics • WebUI 	<p>The Tungsten Fabric Core project intends to be an umbrella project for all TF activities related to the mentioned modules. Goals are defined within the Release process for each Release based on registered blueprints (see the scope of particular release).</p> <p>https://wiki.tungsten.io/x/soVdAQ</p> <ul style="list-style-type: none"> • Core Team: As Umbrella project represents a larger functionality. A core team of Subject Matter Experts with representation from each module will be formed. This team will be responsible for providing feedback to PTL enabling exhaustive review of proposed blueprints in this Project <ul style="list-style-type: none"> ◦ Chandra Mohan ◦ Yuvaraja Mariappan ◦ One committer representation from each module • Committers, eligible to approve blueprints collectively (at least 2 approvers are needed and no negative feedback from the others) <ul style="list-style-type: none"> ◦ Chandra Mohan ◦ Yuvaraja Mariappan ◦ Other Committers are inducted from the following modules: <ul style="list-style-type: none"> ▪ Configuration System ▪ Fabric Management ▪ Control Plane ▪ Data Plane ▪ Kubernetes Orchestrator ▪ Openstack Orchestrator ▪ Deployment ▪ Packaging ▪ Analytics ▪ WebUI
TF Operator Framework	Szymon Krasuski	<ul style="list-style-type: none"> • Operator 	<p>The Tungsten Fabric Operator project intends to provide functionality around Life Cycle Management of Tungsten Fabric deployment. Where, Tungsten Fabric operator will be extending functionalities like ease of deployment, seamless upgrades and auto scaling.</p> <p>Core team:</p> <ul style="list-style-type: none"> • Michael Henkel • Sanju Abraham
Supporting projects			
CI/CD	Andrey Pavlov	<ul style="list-style-type: none"> • Jenkins • dev tools 	TF CI (Jenkins) + devtools (tf-dev-env, tf-devstack, tf-dev-test)

Documentation & Training	Nick Davey	<ul style="list-style-type: none"> • Docs • OpenLab 	<p>The goal of this project is to provide documentation for Tungsten Fabric open source project.</p> <p>The documentation should cover topics as:</p> <ul style="list-style-type: none"> • Why Tungsten Fabrics (TF) exist? • How to install TF? • How to deploy TF? • How to use TF? • How to contribute code to TF? • How to contribute documentation to TF? • What is the governance structure for this open source project? • What are the internal processes and workflows in this open source project (for example CI)? <p>Initial Core Team</p> <ul style="list-style-type: none"> • Nick Davey • Rasika Subramanian
Supporting projects			
Deployment & Packaging	Alexandre Levine	<ul style="list-style-type: none"> • Deployment • Packaging 	<p>Containers and deployers layers for TF. Includes:</p> <ul style="list-style-type: none"> • Containers <ul style="list-style-type: none"> ◦ Docker ◦ Podman ◦ Cri-o ◦ Containerd • Deployers <ul style="list-style-type: none"> ◦ RHOSP (13, 16) - openstack, vrouter-less / RHEL 7, 8 ◦ JuJu - openstack, k8s, hybrid / CentOS 7, Ubuntu 16, 18 ◦ Ansible deployer - openstack, k8s / Centos 7, Ubuntu 16, 18 ◦ Helm - openstack, k8s / Centos 7, Ubuntu 16, 18 ◦ k8s manifests - k8s / Centos 7, Ubuntu 18 ◦ Openshift (3, 4) - RHEL 7, 8 ◦ Future: Operator framework • Integration <ul style="list-style-type: none"> ◦ Akraio / Airship ◦ OPNFV
Security Vulnerability	vacant		

Withdrawn or frozen

previous proposals - replaced by TF Core project during the discussion.

Project	PTL contact	Modules	Short Description
Controller projects			
Configuration	Nagendra Prasath Maynattamai Prem Chandran Ignatious Johnson christopher	<ul style="list-style-type: none"> • Config 	<input type="checkbox"/> Ignatious Johnson christopher- please provide a short description here 06 Aug 2020
Fabric & Device Management	Atul Moghe @Ankur Tandon	<ul style="list-style-type: none"> • Config • Fabric Management • WebUI 	<input type="checkbox"/> Atul Moghe - please provide a short description here 06 Aug 2020
Orchestration	Prasad Miriyala Sachchidanand Vaidya	<ul style="list-style-type: none"> • Kubernetes • OpenStack 	Tungsten Fabric integration with supported Orchestration systems (Kubernetes, OpenStack)
Control Plane (Controller)	Mahesh Sivakumar Nikhil Bansal	<ul style="list-style-type: none"> • Config • Control • DataPlane • WebUI 	<p>The general purpose is to deliver Tungsten Fabric Controller functionality</p> <p>A set of software services that maintain a model of networks and network policies, typically running on several servers for high availability.</p> <p>The Tungsten Fabric controller integrates with cloud management systems such as OpenStack or Kubernetes. Its function is to ensure that when a virtual machine (VM) or container is created, it is provided with network connectivity according to the network and security policies specified in the controller or orchestrator.</p>

Collection and Analytics	Nikhil Bansal	<ul style="list-style-type: none"> • Config • Analytics 	<input type="checkbox"/> Nikhil Bansal - please check and update a short description here 06 Aug 2020 TF monitoring Project. Tungsten Fabric collects information from the cloud infrastructure (compute, network and storage) and the workloads running on it in order to facilitate operational monitoring, troubleshooting and capacity planning. The data is collected in a variety of formats such as syslogs, structured messages (known as Sandesh), lpx, Sflow and SNMP. Objects such as vRouters, physical hosts, virtual machines, interfaces, virtual networks and policies are modeled as User Visible Entities (UVEs) and the attributes for a UVE may come from a variety of sources in different formats.
WebUI	vacant	<ul style="list-style-type: none"> • WebUI 	Need to identify a community representative
Dataplane (vRouter projects)			
vRouter (Dataplane)	Anand Rao Kiran KN		<input type="checkbox"/> Anand Rao, Kiran KN - please check and update a short description here. Please check if DPDK, SR-IOV, Smart NIC projects are included in the scope of this one or should still exist as separate projects. Please update table and description accordingly 06 Aug 2020 The default deployment option today is for the vRouter forwarder to be implemented in a module that runs in the Linux kernel. The vRouter implements networking functionality that would otherwise be performed using iptables or Open vSwitch. Running in the kernel gives the forwarder direct access to network traffic as it passes through the network stack of KVM, and provides a significant performance improvement over what can be achieved if the forwarder ran as a process in userspace. 1) Flow setup rate improvements: The flow setup rate is one of the bottlenecks currently and we are limited to around 40-50K flows per second. There is a need to optimize this to achieve the setup rate in the order of 500-1000K flows/sec. Scope: Medium The Data Plane Development Kit (DPDK) , from Intel, is a set of libraries and drivers that allow applications running in user space to have direct access to a NIC without going through the KVM network stack. A version of the vRouter forwarder is available that runs in user space and supports DPDK. The DPDK vRouter provides accelerated packet throughput compared to the kernel module with unmodified VMs, and even better performance can be achieved if the guest VMs also have DPDK enabled. The DPDK vRouter works by dedicating CPU cores to packet forwarding which loop continuously waiting for packets. Not only are these cores not available for running guest VMs, as they are running at 100% continuously, and this can be an issue in some environments. 1) QoS support for DPDK vRouter: TF vRouter supports QoS marking and enforcement. The marking part is done leveraging Linux kernel support and enforcement in the physical interface is done using a utility called 'qosmap' which configures the NIC queues with the desired scheduling and priority. However, in DPDK, this support is not there. It supports some basic marking abilities, but nothing more than that. We need full QoS marking and enforcement support for DPDK vRouter also. Scope: Medium 2) IPSec support for DPDK vRouter: TF vRouter currently supports vRouter to vRouter encryption using strongswan in kernel mode. We need to have similar support in DPDK mode which leverages DPDK library APIs for IPSec. Scope: Large SR-IOV (Single Root – Input/Output Virtualization) isn't strictly a deployment option for vRouter itself, but can be used with vRouter in some applications. SR-IOV allows the hardware resources of a NIC to be shared among multiple clients as if each has sole access, much like a hypervisor does for CPU. It gives a VM interface direct access to the NIC, so the data path bypasses the hypervisor networking stack, which leads to enhanced performance. SR-IOV can be useful when the VM is performing a gateway function between a physical network and virtual networks, but since SR-IOV involves bypassing the vRouter, the interfaces don't participate in Tungsten Fabric virtual networks and don't participate in network policies and network services. Smart NIC Some new NICs are becoming available which are programmable. The Tungsten Fabric vRouter forwarder functionality can be implemented on these new NICs, and this brings substantial benefits in performance, particularly for small packet sizes which are dominant in some environments. Additionally, forwarding is almost completely offloaded from the x86 CPU of the server, so cores can be freed up for more VMs. Smart NICs look very promising, but obviously require that the Smart NICs are available in production environments, and it will take time for them to become in widespread use.
vRouter Agent	Pranavadatta DN (Pranava Dutta) sangarshan pillareddy		