

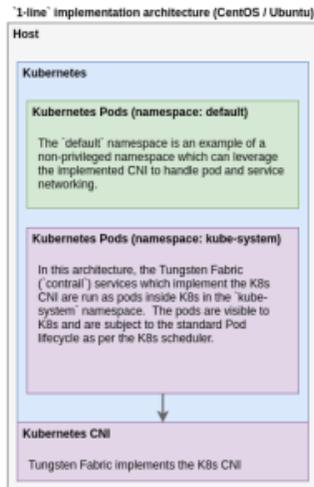
Docs Census

Performed in 2019, the below is a census of what documentation exists, where it lives at the time of the census, and its condition (up to date, out of date, etc).

The most visible entry point for documentation is the "Documentation" navigation entry on <https://tungsten.io/>, which directs the user to <https://tungstenfabric.c.github.io/website/>.

On the <https://tungstenfabric.github.io/website/> site, there are the following components:

- **Getting started with:**
 - **AWS** - This is the 'Carbide' project.
 - **Will Stevens** : I have not validated the state of this documentation.
 - **CentOS**
 - **Resulting Kubernetes CNI Architecture**

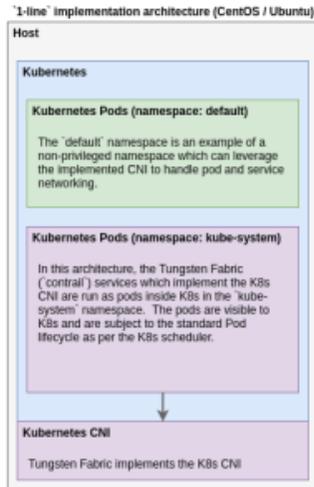


- **Will Stevens** : I have spent quite a bit of time testing this process. However, the [manifest](#) that is used in this process does not seem to be tied to the [currently maintained manifest process](#). It is unclear if the manifests documented in this process is maintained.
- **Will Stevens** : For my own purposes, such as demos and feature validation, I have been maintaining [cloudops/tf-demo](#), which automates the deployment of TF on cloud.ca.

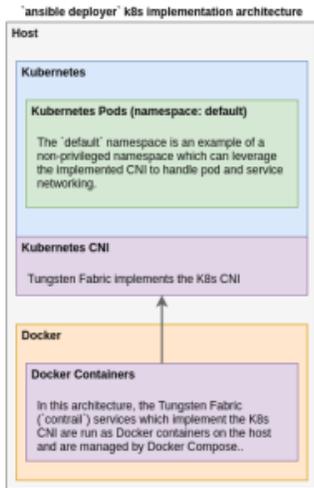
Below I outline a few items which I have struggled with when working with this deployment method:

- The manifests which are [maintained here](#) are different from the [documented manifests](#) for this approach. I had to [generate the official manifests on a specific VM using this process](#) (which unfortunately is specific to a single environment), then I did a manual diff with the documented manifest to [create my own version](#) which I then re-parameterized for my needs. I would like to be able to dynamically build my manifest from a variables file so my manifest is not tracked separately from the official manifests.
 - I have been using this setup to [illustrate cross-region connectivity between two independent K8s clusters](#). I do this by creating two different k8s clusters using this deployer and then BGP peer the two environments together. In order to do this, I needed to ensure that the IP space used for 'pod', 'service' and 'ip_fabric' are different in the two environments. Unfortunately, Tungsten Fabric does not inherit the 'kubeadm init' parameters of '--pod-network-cidr' and '--service-cidr' (at the time of this writing), so you have to update the config inside TF in order to set the IP ranges to the same as what was used in 'kubeadm init'. Recently, [documentation has been added](#) to describe the [parameters which are available to be set](#). In my case, I had to set the following additional params in the 'KUBERNETES_POD_SUBNETS', 'KUBERNETES_SERVICE_SUBNETS' and 'KUBERNETES_IP_FABRIC_SUBNETS' in the 'kube-manager-config' ConfigMap.
 - By default, the instructions will set the 'VROUTER_GATEWAY' parameter to the 'K8S_MASTER_IP', which unfortunately will not work. Essentially, when the TF comes up it will break the host's networking stack with this config, so you will lose SSH access to the host. In order to fix this, I set the 'VROUTER_GATEWAY' to be the gateway for the network on which the 'K8S_MASTER_IP' IP is part of.
 - In order to change the default password for the web UI (since my installs end up on the public internet), I [needed to specify](#) the 'WEBUI_STATIC_AUTH_PASSWORD' param.
 - Unfortunately, I don't remember the details, but I needed to change one of the default ports for one of the databases deployed by the TF architecture because, by default, two different services were trying to use the same port. I was able to fix this by reviewing the logs 'contrail' logs on the host and then researching the different param options.
- **Ubuntu**

- Resulting Kubernetes CNI Architecture



- Will Stevens : I have not tested this process, but it is very similar to the CentOS process above. Again, it is unclear if the [manifest](#) associated with this recipe is maintained, as it does not seem to be associated with the [currently maintained manifest process](#).
 - OpenStack
 - Will Stevens : I have not validated the state of this documentation. It is based on the [Ansible Deployer](#).
- Deployment option not represented at the top level on this page:
 - Ansible Deployer for Kubernetes
 - Resulting Kubernetes CNI Architecture



- Will Stevens : I have not validated the state of this documentation, but I believe it should be added as a top level deployment option.
 - Gary Greenberg : I am still working to validate the documentation for this deployment option (travel and lack of remote access to my lab has slowed my progress... building VPN to address this moving forward).
 - Darien Hirotsu : Our GSoC intern (Nishita) and I have been working on the Ansible Deployer extensively testing against the bare metal provider. Here are some items we have discovered that might be of interest and that may help understand some of the other deployment options
 - The appropriate branch must be used to deploy the appropriate version of Tungsten Fabric
 - To deploy TF 5.0.1, you must use the 5.0 branch from GitHub of the Ansible playbooks (in our experience, we cannot install 5.1 from the 5.0 playbooks since container names have changed)
 - I would guess this also applies to the existing [currently maintained manifest process](#) (i.e. you must deploy from the appropriate branch)
 - The Ansible deployer does not use the [currently maintained manifest](#)
 - I'm a little concerned this means we may end up with different behavior depending on the path a person takes when building out Tungsten Fabric
 - Instead it uses a docker service command
 - There are a few bugs to work out
 - The example README shows the deployment in the incorrect order
 - The playbooks should be run in the order of: `configure_instances.yml`, `install_k8s.yml`, `install_contrail.yml`
 - There are issues with various versions that may be preventing us from deploying applications
 - Here's an example issue:
 - This Juniper document says Docker version 18.06.0-ce should be used with 5.0.1
 - https://www.juniper.net/documentation/en_US/contrail5.0/information-products/topic-collections/release-notes/topic-126833.html
 - However after using the Ansible deployer, we are ending up with version 18.03.1-ce
- 'yesac' from the TF Slack community wrote the following script to support his Ubuntu deployments using the Ansible Deployer: <https://gist.github.com/csiens/f0457432efc513b44cf5d55ccd1194e7>

- **Summits and Events**
 - New features are introduced to the TF community through use of Blueprints.
 - Blueprints are tracked in Jira here: <https://jira.tungsten.io/projects/TFP> (which will redirect you to one of the blueprints)
 - Blueprints should be associated with a Spec. The spec is a detailed technical document which outlines the technical details associated with the feature/function change.
 - Specs are tracked in this repo: <https://github.com/Juniper/contrail-specs>
 - While the Blueprints track the intent, the actual work is tracked in Jira in the following section: <https://jira.tungsten.io/projects/TFB> (again, this will redirect to a specific issue)
 - Despite the Specs being tracked in Github, all the contribution is expected to be done through Gerrit. This is likely a point of confusion...
 - **The Contribution Process**
 - All the code is managed through Gerrit (either the old or new Gerrit), despite all of the repositories being published on Github.
 - In order to contribute, you must first sign one of the CLAs. The CLA must be associated with the contributors account in Gerrit as a CLA is a gating function for being able to accept contribution.
 - ICLA - Is the 'Individual' contributor license agreement and is used when your organization does not have a CCLA which you can be associated with.
 - CCLA - Is the 'Corporate' contributor license agreement and is used to associate many contributors within an organization to a single legal agreement.
 - Once you have a CLA, you will need to install [git-review](#) to properly format and submit merge requests against the repositories in Gerrit.
 - While somewhat outdated and very OpenContrail specific, the best documentation I have found which describes the contribution process is this: <https://github.com/Juniper/contrail-community-docs/blob/master/Contributor/GettingStarted/getting-started-with-opencontrail-development.md>
 - This document should be updated to reflect Tungsten Fabric and moved into the TF docs domain.
 - VM (Vicky) Brasseur (she/her) has done an initial pass at documenting the contribution process here: <https://jira.tungsten.io/browse/TFB-1449>
- **Additional Documentation Resources**
 - *Currently, the only repository in the new-gerrit.tungsten.io is the `docs` repo. In an effort to understand and document the process of contribution in this new Gerrit environment, we are working with this repo: <https://gerrit.tungsten.io/r/admin/repos/docs>*
 - Tungsten Fabric was provided an export of the Contrail documentation, which has been converted to RST format and is currently in the process of being sanitized prior to being incorporated into the core docs repo (above). It can be found here: <https://github.com/tungstenfabric/tf-transitional-contrail-docs>



Unable to locate Jira server for this macro. It may be due to Application Link configuration.

- A Tungsten Fabric training course is being developed. The content covered in this training course is available here: https://docs.google.com/document/d/1xstx1TRDGtuTeal6xks6_S9mPesCD1XXIXd4wI9zn7E/edit
 - UPDATE 9/4/19: Course (SDN Fundamentals, LFS265) is live here: <https://training.linuxfoundation.org/training/software-defined-networking-fundamentals/>
- A "Carbide Evaluation Guide" (CEG) is being developed privately by a Juniper employee (Dmitri Kalintsev) and the guide is <UPDATE 9/4/19> posted here: <https://tungstenfabric.github.io/website/Carbide/CEG/docs/>
- Through the GSoC program, [Fayaz Akhtar](#), is developing a Quickstart Guide for setting up TF as a CNI for K8s. WIP document is here: https://docs.google.com/document/d/1UaC_H0OPiWU_e7KVhLL3rc29q7ZWWkDvXSe3U5EsSU/edit UPDATE: incorporated into [docs/user/GettingStarted/TFforK8s](#)
- Alexandre Levine has put together a WIP document to outline the Tungsten Fabric CI process, which is available here: https://docs.google.com/document/d/1CMwfu_fwgf8Mlae-1z13XvhWBi3p79PKOMSALf2MGf0/edit#heading=h.m5k7iv7g6h15
- Will Stevens has developed automation to deploy K8s + TF on cloud.ca for various demos based on the CentOS "1-line install", with adaptations to get everything working. The source code is available here: <https://github.com/cloudops/tf-demo>
- Juniper's official Contrail Cloud Deployment Guide (has some info relevant to TF): https://www.juniper.net/documentation/en_US/contrail5.0/information-products/pathway-pages/contrail-cloud-deployment-guide-13.0.pdf
- Juniper's Contrail + Kubernetes documentation: <https://github.com/Juniper/contrail-kubernetes-docs>
- Repo describing the features available per release (ish): <https://github.com/Juniper/contrail-specs>
- The "Kubernetes CNI Architecture" diagrams above were created with drive.draw.io. If you would like to update them, please request access to the google docs file here: https://drive.google.com/file/d/1txY503tMGHc3j_xsMUOwNOGSHmtod6Ae/view?usp=sharing
- Automated tests run by the CI can be found here: <https://github.com/tungstenfabric/tf-test>
- Previously some work was done to consolidate the docs at the time. This spreadsheet is that old list: https://docs.google.com/spreadsheets/d/1FXnqK6t3cmYB0EKSH_xGryKWW1BIUjih6HnJbknpCkw/edit?ts=5d2b70b6#gid=0
- KVM `tf-devstack` install guide: <https://gist.github.com/csiens/386887aab41d6c8588eae3968204487>
- Hartmut Schroeder documented some scaling numbers. While the deployment tool referenced is TripleO, the information should still be useful: <https://github.com/hartmutschroeder/contrailandrhosp10#21sizing-the-controller-nodes-and-vms>
- The wikis attached to the original OpenContrail repositories have useful information for troubleshooting and to get a deeper level understanding of the Tungsten Fabric design/architecture. The majority of the information resides within the controller repository's wiki.
 - <https://github.com/Juniper/contrail-controller/wiki>
- API interaction for automation
 - [Tungctl](#) create tungsten fabric objects from YAML. Uses python to convert the YAML to JSON and post it to the api.
 - [iosonounrouter](#) blog post with good vnc_api python network creation example.
 - [API server documentation](#) (latest version can be found here https://www.juniper.net/documentation/product/en_US/contrail-networking#cat=developer_resources)

- Contrail API CLI <https://contrail-api-cli.readthedocs.io/en/latest/>



Unable to locate Jira server for this macro. It may be due to Application Link configuration.

- [Tony Liu's](#) API Configuration documents with multiple examples for python, REST/json, go, java, and heat.
 - Good "day one" dive into the interaction of Contrail / Tungsten with Kubernetes: https://www.juniper.net/documentation/en_US/day-one-books/day-one-containers-kubernetes-contrail.pdf
- **Work In Progress... Please contribute if you know of documentation which is not represented here...**