

Loom DevX Guide

Release 0.7.0

Veritas Technologies LLC

Apr 07, 2018

CONTENTS

1	Introducing DevX	3
2	DevX Elevator Pitch	5
3	DevX Use Cases	9
3.1	Inter-dependent Micro Service Architecture Integration Testing	9
3.2	New Loom Feature Development or Loom Application Development	9
3.3	Flexibility in choosing the type of test infrastructure	9
3.4	Replicating Test Environments across teams and geographies	10
4	How to use DevX?	11
4.1	DevX Workflow	11
5	DevX Security	13
6	DevX Tutorial Video	15
7	Other DevX References	17

- *Introducing DevX*
- *DevX Elevator Pitch*
- *DevX Use Cases*
 - *Inter-dependent Micro Service Architecture Integration Testing*
 - *New Loom Feature Development or Loom Application Development*
 - *Flexibility in choosing the type of test infrastructure*
 - *Replicating Test Environments across teams and geographies*
- *How to use DevX?*
 - *DevX Workflow*
- *DevX Security*
- *DevX Tutorial Video*
- *Other DevX References*

INTRODUCING DEVX

DevX Tool can help you speed up Loom Integration Testing, Loom Application development and integration with Veritas Products. Key audience comprises of Veritas Product Dev, Service Integration Dev, Micro-service based product development and verification Engineers who need a fast, reliable way to obtain cluster resources on demand and return it to the pool for other Veritas users.

The following slide captures the value proposition of DevX Tool in a single line:

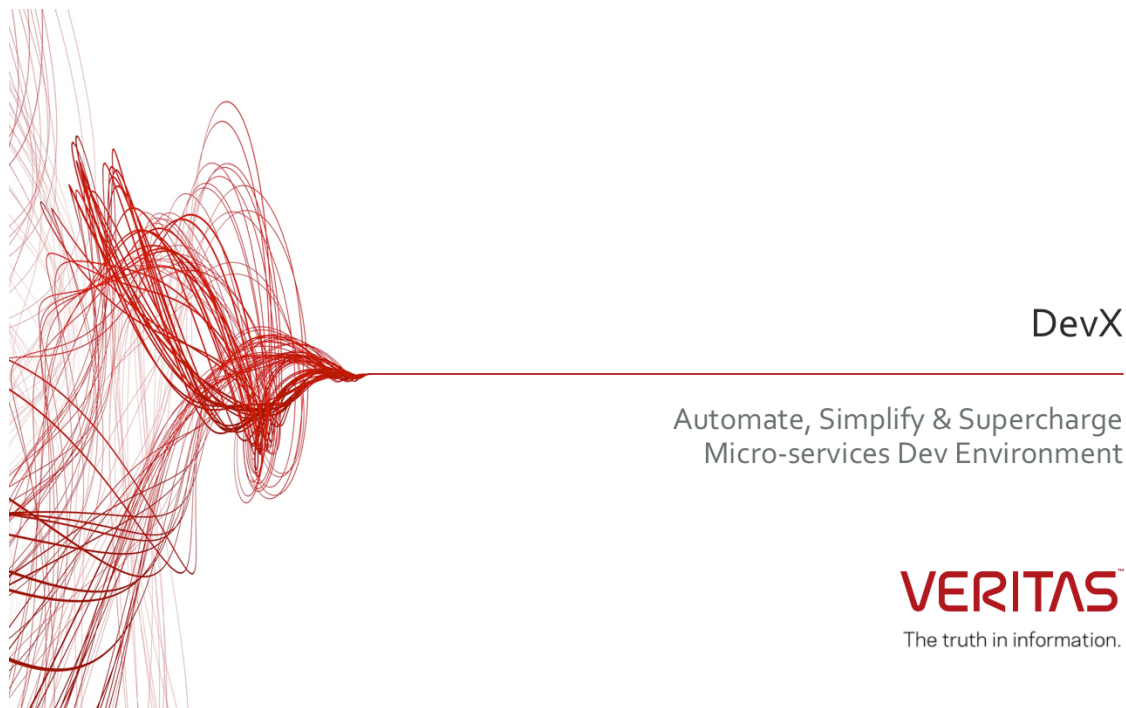


Fig. 1.1: Figure: DevX Value Proposition

DevX provides runtime Kubernetes cluster infrastructure for deploying one or more inter-dependent container based micro-services quickly through REST APIs. DevX supports one micro-service per container setup that is hosted using Kubernetes and defined using Helm, Kube OpenAPI Spec definitions.

DEVX ELEVATOR PITCH

The core value proposition of DevX is automation of micro-services application development environment. The figure below explains at a very high level some of the key features offered by DevX.

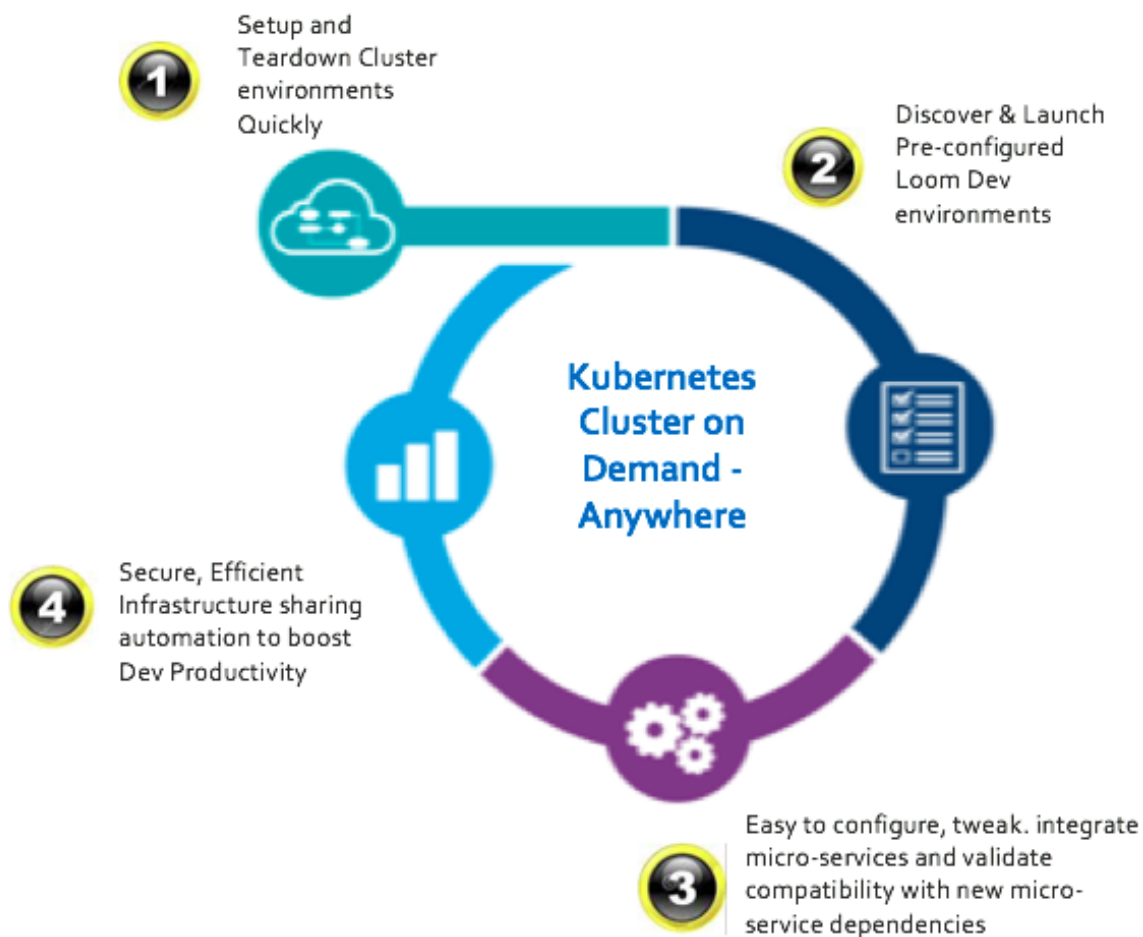


Fig. 2.1: Figure: What is DevX?

The following figure shows the main DevX web interface. In future, there will be a more sophisticated product UI that will be available for this tool.

DevX User Interface

<http://devx-swagger-canary.cicd.veritas.com/>

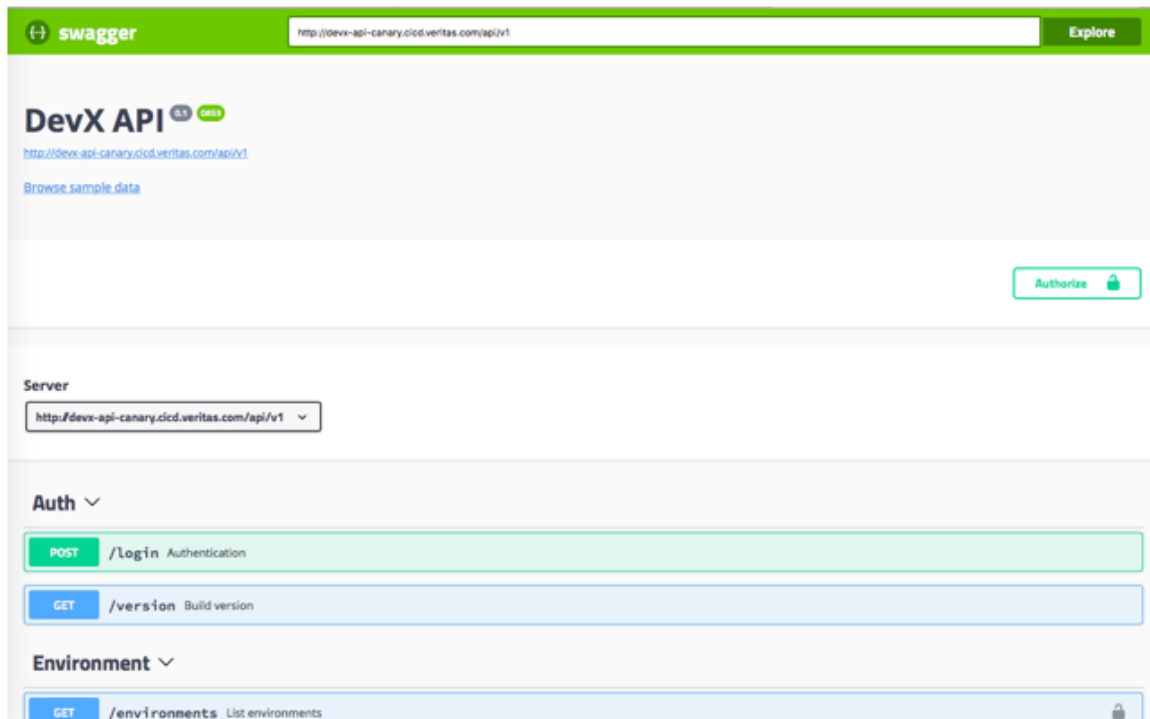


Fig. 2.2: Figure: DevX Web Interface

The key points of DevX Elevator Pitch are highlighted in the figure below:

- DevX is an Internal Veritas tool with REST APIs, aimed at solving developer pain points faced while developing and integrating with Veritas products that have a micro-services based architecture and complex deployment.
- DevX provides Kubernetes Cluster on demand, on any infrastructure resources of developer's choice – be it any cloud provider, physical or virtual servers. Azure, AWS, OpenStack is supported now and others will be available in future.
- Developers can simply focus on creating features or newer applications and leave the deployment, scalability, availability, tearing up and down of environment during integration testing to DevX.
- Developers can use DevX REST API to list available environment templates for Kubernetes Cluster environment discovery, customize these environments to suit their needs or use them as is. Both Helm and Kube Spec based specifications are supported.

Fig. 2.3: Figure: DevX Elevator Pitch

DEVX USE CASES

- Inter-dependent Micro Service Architecture Integration Testing
- New Loom Feature Development or Loom Application Development
- Flexibility in choosing the type of test infrastructure
- Replicating Test Environments across teams and geographies

3.1 Inter-dependent Micro Service Architecture Integration Testing

Micro-services architecture based products are fairly complex. Inter-dependency of these micro-services and variance in the rate of evolution of co-dependent services can cause a significant spurt in integration testing and validation efforts. This also drives up the need for cluster resources, especially when HA, replication and other inter-linked features come into the picture. For products such as Loom, developers often require different versions of inter-dependent micro-services and there is a need to quickly validate a micro-service that is under development, whether it works with one or more newly updated or bug-fixed micro-service dependency. DevX Tool can immensely help in such cases by quickly spinning up test cluster resources and once a developer has completed the use of the cluster, all the resources are automatically returned to the DevX pool for other developers to use. This not only boosts developer productivity but also channelizes and optimizes shared resource usage in complex micro-services cluster environments.

3.2 New Loom Feature Development or Loom Application Development

Loom deployment architecture is fairly complex. There are several moving blocks, each of which themselves are under evolution. In such cases, it can be tricky to identify and obtain a recommended development environment setup. For new feature development or application development, getting a verified and recommended Loom environment, which is approved and known to work, and can be used as a base to enhance one or more micro-services or create a new Loom Application which is deployed on Loom itself, can be a time consuming task, prone to errors. DevX Tool can help you choose the right environment for your specific Loom development needs. You can choose from DevX Environment Directory Listing and Build repository (Artifactory) Listing interfaces and rest assured that you will be able to obtain and spin up your Kubernetes test cluster in a jiffy.

3.3 Flexibility in choosing the type of test infrastructure

DevX offers a generic Kubernetes test cluster on any cloud, VMware based resources on-premises or OpenStack based infrastructure. Each developer need not have to worry about provider specific credentials or deployment specifics.

DevX takes care of that part. Developers can simply use their enterprise SSO credentials to avail DevX API benefits and web based interface to get started with DevX.

3.4 Replicating Test Environments across teams and geographies

DevX ,in future, will also enable users to snapshot complex deployment environments and replicate them elsewhere. This can significantly speed up and ease replication of test environments during testing and production phases or during shared development of micro-services based products and Applications using such products or deployed on a micro-services platform architecture.

The following figure highlights benefits of DevX Tool:

DevX Benefits

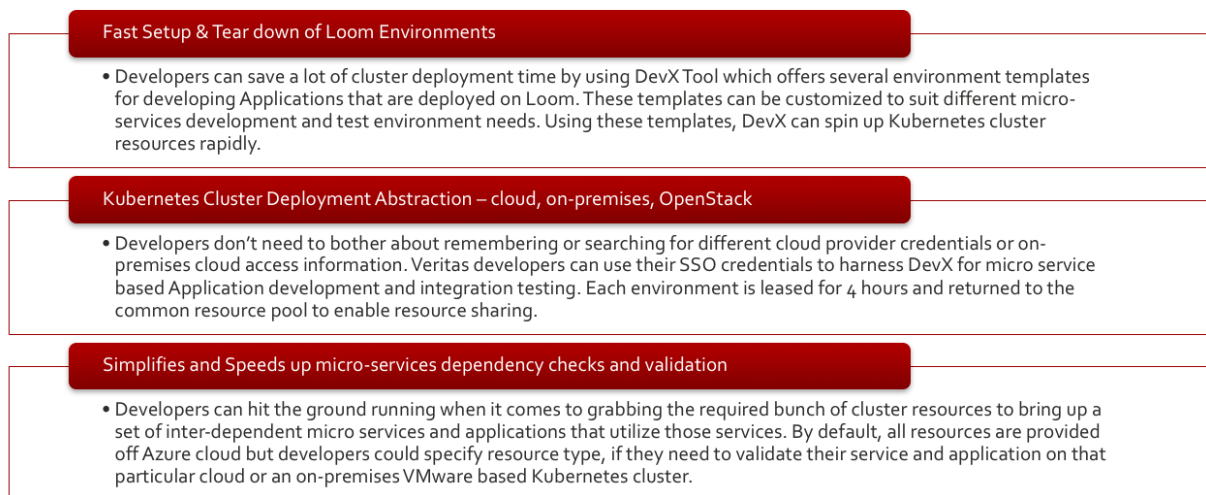


Fig. 3.1: Figure: DevX Benefits

HOW TO USE DEVX?

DevX Tool provides REST APIs for all its functionality of Kubernetes Cluster Abstraction. Refer to [DevX Canary Site](#) for details and list of DevX APIs available for use.

4.1 DevX Workflow

Following are the basic instructions to get started with DevX:

1. Step 1: **Authenticate**

Before you can get started with DevX, you need to authenticate yourself using your Veritas SSO credentials at the [DevX Canary Site](#). You can use this web interface directly to authenticate yourself.

1. Step 2: **List available DevX Environment Definitions**

Next, you can obtain a list of available DevX environments and use one of them to setup your cluster. Alternatively you can create (and delete) your own cluster environment using DevX APIs.

1. Step 3: **List available builds to be deployed**

To help developers identify which Loom or other micro-service builds are available for deployment and their version numbers, DevX provides Artifactory API which acts as a directory of builds that can be deployed for application development and testing.

1. Step 4: **Execute**

Once you have specified the environment and which builds to deploy, you can Execute DevX Environment of your choice – using its ID or ??? (List environment API returns ID, right?). TBD - How does one ‘activate’ the cluster and access it (by name?) – how do you get its IP address? TBD

1. Step 5: **Scale up on down on demand:**

Scale DevX Environment of your choice – Replication factor, per microservice, load balancing and controller functionality are abstracted internally. How to??

DEVX SECURITY

DevX and Kubernetes APIs authenticate users who present credentials, and then authorize them based on their Veritas SSO and role. Developers typically make REST API calls from web interface via their browsers. Every container that runs on the cluster is associated with a service account. This account is also used to list and pull,push images, builds and deployment components.

DEVX TUTORIAL VIDEO

The following is a placeholder video that shows a typical DevX usage flow. We plan to provide specific short videos to describe how to do specific tasks using DevX in subsequent revisions.

Click on this link to see sample [DevX Tutorial Video](#).

OTHER DEVX REFERENCES

TBD