DEFINITION

single-tenancy

Alexander S. Gillis, Technical Writer and Editor

Single-tenancy is an architecture in which a single instance of a software application and supporting infrastructure serves one customer. Single-tenancy is commonly implemented in software-as-a-service (SaaS) delivery models or in cloud services. In single-tenancy architectures, a customer -- called a tenant -- will have a singular instance of a SaaS application dedicated to them.

In a single-tenant architecture, the host provider will aid in managing the software instance and dedicated infrastructure while still lending nearly full control to a single tenant for customization of software and infrastructure. Some common characteristics of single-tenancy models are that they tend to provide a high level of user engagement and user control, as well as reliability, security and backup ability. Because tenants are in a separate environment from one another, they are not bound in the same way tenants using a shared infrastructure would be.

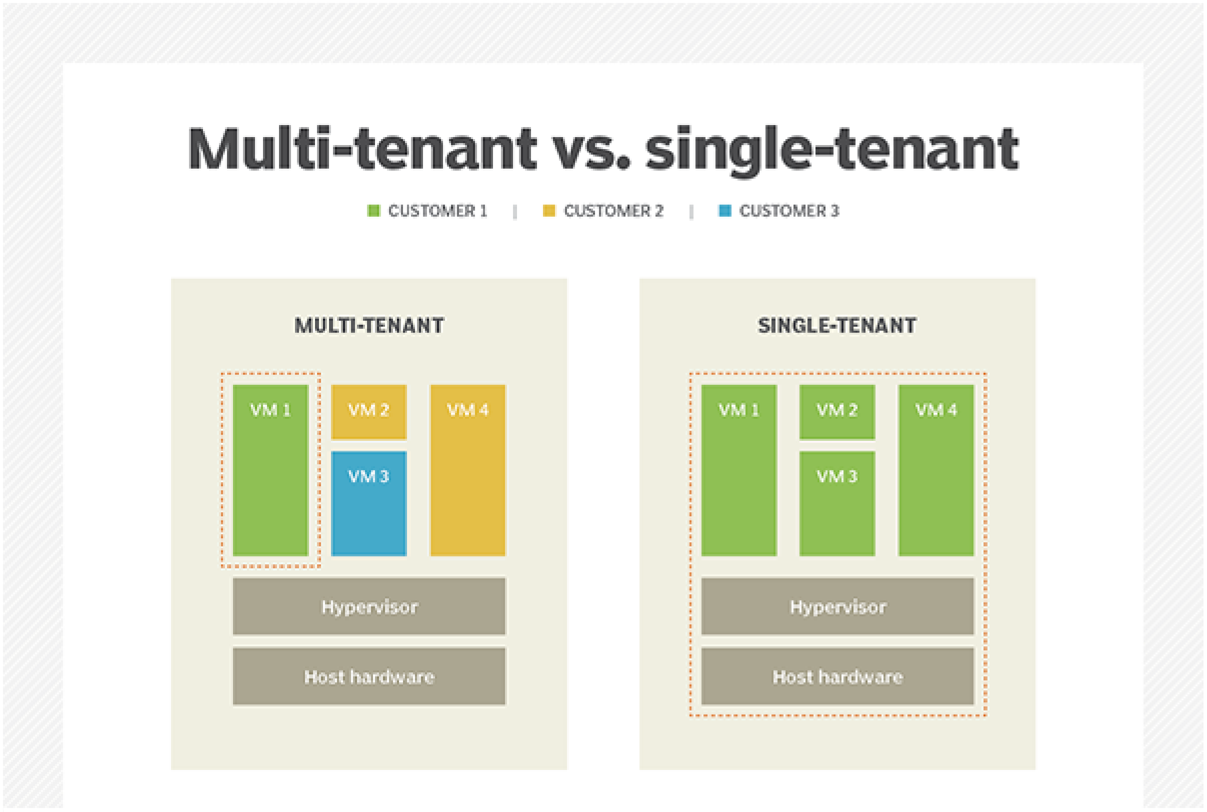
Potential customers would likely choose a single-tenant infrastructure over other possible options for the ability to have more control and flexibility in their environment for addressing specific requirements.

How single-tenancy works

In a single-tenancy architecture, every tenant will have their own single database and software instance. This way, each tenant's data is isolated from one another. In addition, the architecture is designed to only allow one instance per SaaS server. Each piece of software may be purpose-built for the new tenant, or the tenant can customize the user interface (UI) after installation. Once the software is installed locally, tenants can typically customize the software to best suit what is needed for their specific environment, but they do not have access to any underlying code.

Each tenant's data should also have an isolated backup, so if there is any data loss, tenants should have an easy time restoring their data. In addition, tenants can typically choose when to install any available updates individually, instead of waiting for the service provider to do so.

Cloud adoption of single-tenancy architectures in cloud computing is common as well. In most cases, if someone is using a private cloud service or a third-party cloud offering, it is most likely a single-tenant system. This is because an individual would be the only customer with access to that instance, with security and management options as well as individual controls.



A comparison of multi-tenancy and single-tenancy architectures per customer.

Benefits of single-tenancy

Although single-tenancy architecture is used less often, it still has some noticeable benefits that keep it open as an option when deciding on a service architecture. Some benefits include:

- Single-tenant instances through a cloud service or SaaS.
- Data is independent of other potential tenants with the same provider.
- Data security. Even if there is a data breach to one tenant with the same service provider, another tenant would be safe from the breach since data is stored in a separate instance.
- Since all of a customer's data is separate, a large degree of customization is possible for software and hardware instances.
- Single-tenant instances are considered reliable, since performance is based on only one instance, instead of many from different tenants.
- Restore and disaster recovery. Isolated backups allow users to quickly enable a recovery if there is a disaster and data is lost.
- Single-tenancy may also lend itself to migrating from a host environment if needed.

Drawbacks of single-tenancy

With all the potential advantages to single-tenancy, it is still the less used option out of competing architectures, which could be due to some of its disadvantages. Drawbacks to single-tenancy include:

- Between setup time, resources, customization, and maintenance, hosting one SaaS instance per customer can come at a price.
- Since the tenant is normally the one to manage a single-tenant system, it then takes more time to update, upgrade or manage something.
- Learning curves may appear when first beginning to implement and customize a single-tenancy SaaS.
- In a less optimized system, not all resources may be utilized, which makes for a less efficient system.

Requirements for single-tenancy

Requirements needed in single-tenant environments include:

- If in a SaaS product or a cloud service, the service should be able to meet with whatever the requirements are for future workloads.
- Initial startup time. Single-tenancy involves significant startup time since the software has to be built or customized for each tenant.
- Resources for maintenance. Single-tenant environments tend to require more maintenance and upkeep, meaning end users should have the resources and time needed for the upkeep.
- Since the underlying code of a single-tenancy SaaS application is blocked off, major extensions and third-party integrations may require administrative support from the host service.

Multi-tenancy vs. single-tenancy

Single-tenancy is typically contrasted with Multi-tenancy, an architecture in which a single instance of a software application serves multiple customers. In a multi-tenant architecture, each customer shares the same database and application. Multi-tenancy is typically ideal for businesses that want an easier startup experience and fewer hardware requirements. The architecture has become an industry standard for enterprise SaaS environments. In comparison to single-tenancy, multi-tenancy is cheaper, has efficient resource usage, has a lower maintenance cost and a potentially larger computing capacity.

among other tenants. In addition, multi-tenancy can experience more downtime.

When deciding between single- and multi-tenancy, users should consider how much customizability is wanted or needed. However, it is important to note that most SaaS services operate on multi-tenancy.

This was last updated in January 2020

Continue Reading About single-tenancy

- How to plot out an Office 365 tenant-to-tenant migration
- Do multi-tenant environments still create noisy neighbors?
- Office 365 multiple tenants vs. single tenant considerations
- Kubernetes security opens a new frontier: Multi-tenancy
- Understanding the anatomy of a UCaaS platform

Related Terms

Amazon Lightsail

Amazon Lightsail is an Amazon cloud service that offers bundles of cloud compute power and memory for new or less experienced ... See complete definition

cloud engineer

A cloud engineer is an IT professional responsible for any technological duties associated with cloud computing, including design... See complete definition

What is a private cloud?

Private cloud is a type of cloud computing that delivers similar advantages to public cloud, including scalability and ... See complete definition

Dig Deeper on Cloud architecture design and planning

Epicor Software applications make transition to SaaS

By: **Jim O'Donnell**

Software as a Service (SaaS)

By: Wesley Chai

SaaS ERP vs. cloud ERP: What's the difference?

By: Lindsay Moore

SaaS ERP

By: David Essex

SERVER VIRTUALIZATION VMWARE VIRTUAL DESKTOP AWS DATA CENTER WINDOWS SERVER

SearchServerVirtualization

Get the right components for your home lab VM

Home lab VM setups are helpful to test technology and build VM management skills. Be sure to evaluate RAM, CPU, network speeds ...

A beginner's guide to a multistage Docker build

Multistage builds help you manage Dockerfile size during container proliferation. Get the process right with the necessary ...

Guides Opinions Photo Stories Quizzes Tips Tutorials Videos