Different types of infrastructure provisioning methods, each providing varying degrees of control, automation, and connection requirements for installing or deploying environments.

A close-up of a diagram

Description automatically generated

1. **Automated - Installer Provisioned Infrastructure**:

• The infrastructure is automatically provisioned, requiring minimal manual intervention.

• It is Kubernetes (KS) related, suggesting orchestration or container management.

• Supports self-service, meaning users can independently manage their environment setups.

1. **Full Control - User Provisioned Infrastructure**:

• Users can bring their own hosts (computers or servers) for deployment.

• You have the flexibility to choose and control your infrastructure automation, offering more customizability.

• It provides full flexibility, allowing for deep customization and integration of Independent Software Vendor (ISV) solutions.

1. **Interactive – Connected - Assisted Installer**:

• This method involves a web-based, guided installation experience, providing users with step-by-step assistance.

• It is infrastructure agnostic, supporting a variety of environments including bare metal (physical servers), vSphere, and Nutanix.

• The installation process is ISO-driven, meaning it relies on ISO images to install and set up the infrastructure.

1. **Local – Disconnected - Agent-based Installer**:

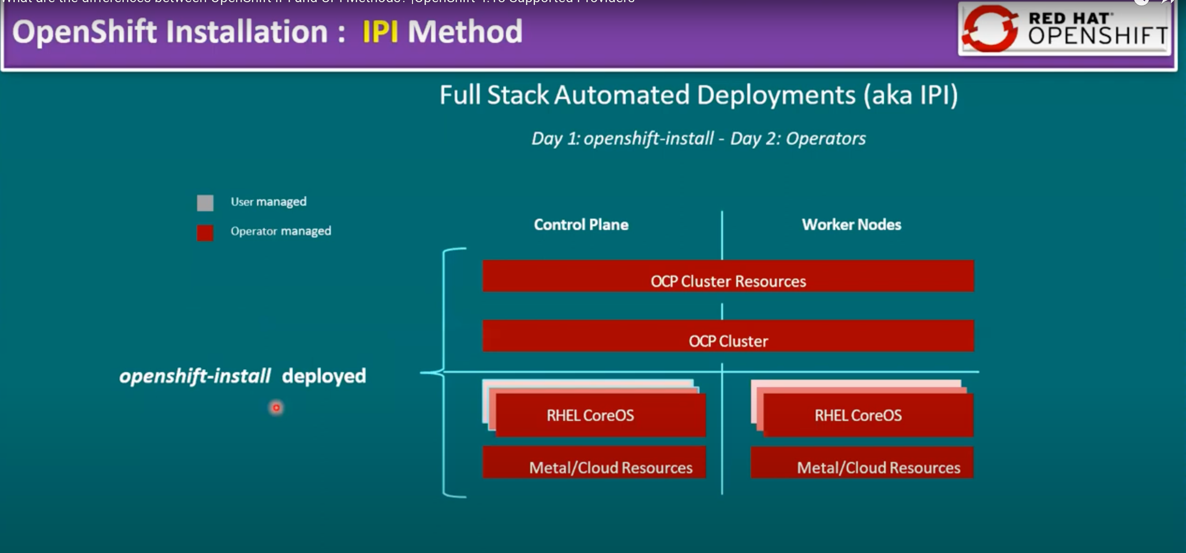
• Designed for environments that are disconnected or “air-gapped,” where no internet access is available.

• The installations can be automated using the Command Line Interface (CLI), which allows for scripting and remote executions.

• Supports installations on bare metal, vSphere, and Single Node OpenShift (SNO).

• ISO-driven for offline installations.

**OpenShift Installation: IPI (Installer Provisioned Infrastructure) Method**



This image explains the OpenShift Installation: IPI (Installer Provisioned Infrastructure) Method, which is used for fully automated deployments of OpenShift clusters.

Key Elements:

1. **Full Stack Automated Deployments (aka IPI)**: This is an automated deployment method where the OpenShift installer provisions the infrastructure and sets up the OpenShift Container Platform (OCP) cluster on Day 1. On Day 2, operators take over to manage the cluster and resources continuously.
2. **Control Plane and Worker Nodes**: The diagram shows two primary components: the Control Plane and the Worker Nodes. These are the core building blocks of a Kubernetes/OpenShift cluster.

* Control Plane: It contains the OCP Cluster Resources, which manage the overall cluster state and ensure proper coordination between nodes. These resources are operator-managed, meaning the cluster is controlled by software operators that automate tasks like scaling, updates, and maintenance.
* Worker Nodes: These are the actual servers or instances where application workloads run. Like the Control Plane, the worker nodes are also operator-managed, meaning most of their tasks are automated by the system.

3. **Managed Components:** Red Color (Operator Managed): Both the control plane and worker nodes are managed by operators. Operators are built-in automation tools that handle complex operational tasks for Kubernetes/OpenShift resources.

* **RHEL CoreOS**: The underlying operating system for both control plane and worker nodes is Red Hat Enterprise Linux CoreOS (RHEL CoreOS), which is optimized for OpenShift.
* **Metal/Cloud Resources**: The infrastructure (bare metal or cloud-based resources) is used to deploy both control plane and worker nodes. These resources are provisioned and managed by the OpenShift installer.

**OpenShift Installation: UPI (User Provisioned Infrastructure) Method**

A screenshot of a computer program

Description automatically generated

**Key Elements:**

1. **Deploying to Pre-existing Infrastructure (UPI)**: Unlike the IPI method, UPI involves **customer-managed** infrastructure. The deployment uses existing infrastructure resources (such as servers or cloud environments) that the customer has already provisioned.
2. **Control Plane and Worker Nodes**: Like the IPI method, the cluster is divided into a **Control Plane** and **Worker Nodes**.

* **Control Plane**: The control plane includes the **OCP Cluster Resources** and the core **OCP Cluster**, which are operator-managed. The control plane must run on **RHEL CoreOS** (a requirement for this part of the infrastructure), and these nodes are managed by operators to ensure smooth operations.
* **Worker Nodes**: The worker nodes can run on either **RHEL CoreOS** or **RHEL 8/9**, providing more flexibility depending on the user’s preference. The worker nodes can also be provisioned on customer-managed infrastructure, whether it’s bare metal or cloud-based resources.

1. **Managed Components**:

* **Red Color (Operator Managed)**: Operator-managed components include the OCP Cluster Resources and the OpenShift cluster itself. These parts are managed by operators that handle automated maintenance, updates, and scaling.
* **Gray Color (User Managed)**: The **metal/cloud resources** (infrastructure) and some worker node management tasks are handled by the user or their system administrators. Unlike IPI, the customer is responsible for provisioning and maintaining these underlying resources.

It is important to note that **Control Plane nodes must run RHEL CoreOS**, meaning these nodes cannot use other operating systems for proper compatibility and operation within the cluster.

**Comparison between two OpenShift deployment methods: Full Stack Automation (IPI) and Pre-existing Infrastructure (UPI)**

|  |  |  |
| --- | --- | --- |
| **Task** | **Full Stack Automation (IPI)** | **Pre-existing Infrastructure (UPI)** |
| **Build Network** | Installer | User |
| **Setup Load Balancers** | Installer | User |
| **Configure DNS** | Installer | User |
| **Hardware/VM Provisioning** | Installer | User |
| **OS Installation** | Installer | User |
| **Generate Ignition Configs** | Installer | User |
| **OS Support** | RHEL CoreOS | RHEL CoreOS + RHEL 8/9 (Worker nodes) |
| **Proxy Support** | No | Yes |
| **Disconnected Installation Support** | No | Yes |
| **Private Cluster Support** | Yes | No |

1. **Build Network**: This refers to the process of setting up the network infrastructure, which involves configuring network components like IP addressing, subnets, routing, and firewalls to ensure that nodes can communicate with each other within the OpenShift cluster.
2. **Setup Load Balancers**: Load balancers distribute incoming traffic across multiple nodes in a cluster. Setting them up ensures that the traffic is evenly distributed, preventing any one server from being overwhelmed, and improving the availability and scalability of applications.
3. **Configure DNS**: DNS (Domain Name System) configuration allows the OpenShift cluster and its services to resolve domain names to IP addresses. This is necessary for internal communication within the cluster, as well as for external services to access the cluster and its applications.
4. **Hardware/VM Provisioning**: Provisioning refers to the process of setting up physical hardware or virtual machines (VMs) to serve as nodes in the OpenShift cluster. This includes allocating compute resources, storage, and network interfaces required to run the cluster.
5. **OS Installation**: Installing the operating system (OS) on the nodes, typically **RHEL CoreOS** in OpenShift. The OS must be properly installed and configured to run the OpenShift components and support the workloads in the cluster.
6. **Generate Ignition Configs**: Ignition configuration files contain machine-specific configurations required to bootstrap and configure nodes during the initialization of the OpenShift cluster. These files include settings such as networking, storage, and user accounts.
7. **OS Support**: This refers to the ongoing maintenance and management of the operating system used by OpenShift nodes, including updating, securing, and ensuring compatibility with OpenShift components. RHEL CoreOS is the primary OS used for control planes, while worker nodes may support other versions of RHEL.
8. **Node Provisioning / Autoscaling**: Node provisioning is the process of adding new nodes (machines) to the OpenShift cluster to handle more workloads. **Autoscaling** automatically increases or decreases the number of nodes based on the demand or resource usage within the cluster.
9. **Customization & Provider Support**: This refers to the ability to customize the OpenShift environment to fit specific infrastructure needs, and which cloud or hardware providers (like AWS, Azure, GCP, Bare Metal) are supported for the deployment.
10. **Proxy Support**: Proxy support allows traffic to pass through a proxy server, which can be important in environments with strict network security policies. UPI supports this feature, while IPI does not.
11. **Disconnected Installation Support**: This refers to the ability to install OpenShift in environments that are completely disconnected from the internet (air-gapped environments). UPI supports disconnected installation, allowing users to manually provide the necessary resources without internet access. IPI does not support this feature.
12. **Private Cluster Support**: Private cluster support allows for the creation of clusters that are isolated from the public internet, typically used for internal or private deployments. This feature is available in IPI, enabling automated creation of such clusters. UPI does not provide this option automatically, as the user must handle the isolation manually.
13. **Existing Virtual Private Networks (VPN)**: This feature allows the integration of OpenShift with existing Virtual Private Networks (VPNs) to extend secure connectivity across different environments. IPI supports this feature, automatically configuring the integration. UPI, on the other hand, requires users to manually configure the integration with any existing VPNs.