# Rehydration keeps systems up-to-date

So far, you've learned that you can use either patching or rehydration to maintain your network systems and conduct updates for vulnerability remediation. Recall that patching helps you maintain your cloud system by letting you add both updates for improving stability and new features. And remember that rehydration is a more recent approach to updating systems and applications in the cloud that involves creating new servers with the latest updates patches, and then decommissioning or destroying the outdated servers.

In this reading, you'll learn more about the best practices you can follow to identify vulnerabilities and assess impact and mitigation strategies using patching and rehydration.

## Enterprise patch management best practices

A cloud security team can use the enterprise patch management steps to identify, prioritize, acquire, install, and verify the installation of upgrades, patches, and updates throughout an organization's systems. Let's review these steps:

1. Develop a cloud-based inventory and baseline.
2. Identify vulnerabilities and available patches for the system.
3. Assess the patch's impact on the system.
4. Download and test the patch in a non-production environment.
5. Schedule the downtime or plan a service restart.
6. Backup data and system configurations.
7. Apply the patch to the system.
8. Verify the patch's effectiveness and stability.

## Patch management planning and principles

To help ensure your patches go well, it's important for you to know about patch management planning and principles.

### Patch management planning

To begin patch management planning, you first need to develop a patching schedule to strike a balance between security and performance. Applying patches too often can disrupt network performance, but waiting too long to apply patches can leave your network vulnerable to attack.

Second, you need to test patches before deploying them. Before applying a patch to your cloud-based networking devices, test it in a staging environment to ensure that it doesn't cause any problems.

Finally, you need to have a rollback plan in place. In case a patch causes problems, have a plan to roll back to the previous version of the software.

### Patch management principles

The U.S. National Institute of Standards and Technology has principles it recommends cloud security teams strive for, including:

- Prepare for problems, since they're inevitable.
- Simplify decision-making.
- Use automation for patching and in emergency situations to give your organization scalability and agility when responding to risks.

## Rehydration best practices

Like patching, rehydration helps you maintain the critical infrastructure of your cloud system and lets you and your users store, access, and manage data. While you need to be able to access your data efficiently and securely, you need to reduce as much risk as possible.

A cloud security team can use the rehydration management steps to follow a continual replacement with up-to-date instances throughout an organization's systems. Let's review these steps:

1. Inventory all resources.
2. Identify vulnerabilities.
3. Run updated and old instances to compare.
4. Redirect traffic from old to new instances.
5. Decommission or destroy old instances.

### Rehydration management planning

To help ensure your rehydration mechanics go well, it's important that you know about rehydration management planning, principles, and rehydration benefits over patching.

One of the great benefits to rehydration is in the context of immutable infrastructure, which promotes unmodified components after deployment and reduces risks. You can use rehydration with many servers at a time and enable DevOps engineers to scale many rehydrations at the same time.

Unlike patching, rehydration is not about regular updates or fixes. Instead, rehydration is continual. To conduct effective rehydration management you need to regularly:

- Ensure devices have sufficient resources.
- Focus on asset availability and performance.
- Only update systems as needed to provide a minimal impact or downtime on user access to assets.

### Rehydration management principles

Here are a few principles cloud security teams should aim to follow:

- Monitor cloud-based networking devices and usage of cloud-based networking devices to ensure that they have the resources they need to operate properly.
- Scale resources for cloud-based networking devices to meet the demand when needed.
- Use a cloud-based monitoring solution to help monitor the resource usage and performance of cloud-based networking devices.

### Rehydration benefits over patching

Using only patching to find and patch all the problems in the system will take time. You need to identify all your vulnerabilities, monitor and test all the patches, and deploy those patches. When you rehydrate, you create an already updated server, so you don't have to look for and apply individual patches. A patch can fail and create problems. When you rehydrate, everything is already patched and ready.

## Patching and rehydration considerations

When you patch or rehydrate your infrastructure, consider these best practices:

- Before you update your infrastructure, plan ahead and make sure you communicate your plans to stakeholders.
- While you're updating, make sure to document any changes you make to your infrastructure in real time.
- After you finish the updates, test your infratructure's performance and consider if there are any issues that could be problems in the future.

**Pro tip:** Rehydration has benefits for cloud computing, but there are instances where patching is also necessary. During your cloud security career, you'll likely use a combination of both patching and rehydration.

## Key takeaways

As a cloud security professional, there are best practices to follow when you need to patch or rehydrate your infrastructure. You can use patch management to identify, prioritize, acquire, install, and verify the installation of upgrades, patches, and updates. You can use the cloud-native approach of rehydration to focus on continual replacement with up-to-date instances. In the cloud, rehydration is the more common approach in the context of immutable infrastructure. You'll likely be using both patching and rehydration, but rehydration can help you save time and avoid glitches from failing patches.