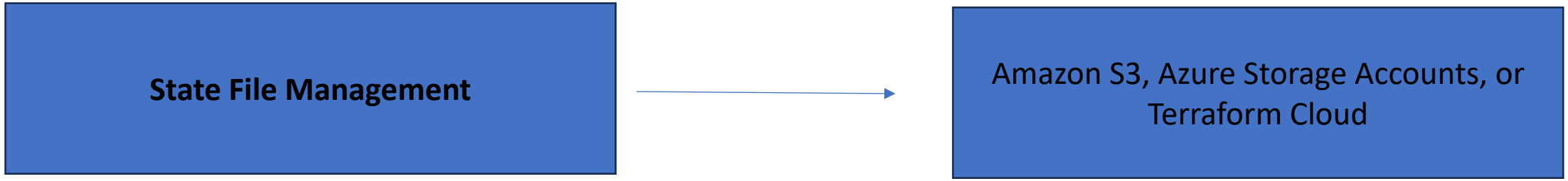
Terraform Deployment Options

State & Code File Management

State File Management: To manage Terraform state files securely and In collaboration, it’s recommended to use remote state management solutions like Amazon S3, Azure Storage Accounts, or Terraform Cloud.



Code Management: For storing and managing our infrastructure code, it's essential to use version control systems such as Bitbucket, GitHub, or Azure Repos.

Optimizing Terraform Execution and Deployment

In terms of code implementation, which refers to executing the code, here's what we have done so far:

- **State management**: Terraform state is stored in an **Azure Storage Account**.
- **Version control**: Code is managed in **Azure Repos**.
- **Execution**: Terraform code has been executed manually via the **command line**.

However, we can enhance this process by implementing CI/CD pipelines using tools like **Azure DevOps**, **Jenkins**, **GitLab**, or **Terraform Cloud** to automate and streamline code execution.

Azure DevOps Implementation

We are going to use Azure DevOps for automating the Terraform deployment. The following steps will be carried out:

- Create a dedicated organzitaion & install the extension
- A dedicated project will be created in Azure DevOps.
- A service connection between Azure DevOps and Azure will be established & provide the SP access at subscription level
- A dedicated repository will be created in Azure DevOps to store the code.
- A storage account will be set up in Azure to store the Terraform state file.
- A YAML file will be created to handle the CI/CD pipeline.
- A variable group will be configured for managing environment-specific variables.

Join us in our Adventure

https://www.linkedin.com/in/akash-kumar-480b3858/

https://www.instagram.com/akash_sinha08/