

# Accelerating Open-Source AI with AWS Bedrock: Architecting LLM Integration with Posit Workbench & Positron

Phil Bowsher, Posit/RStudio PBC

## Abstract

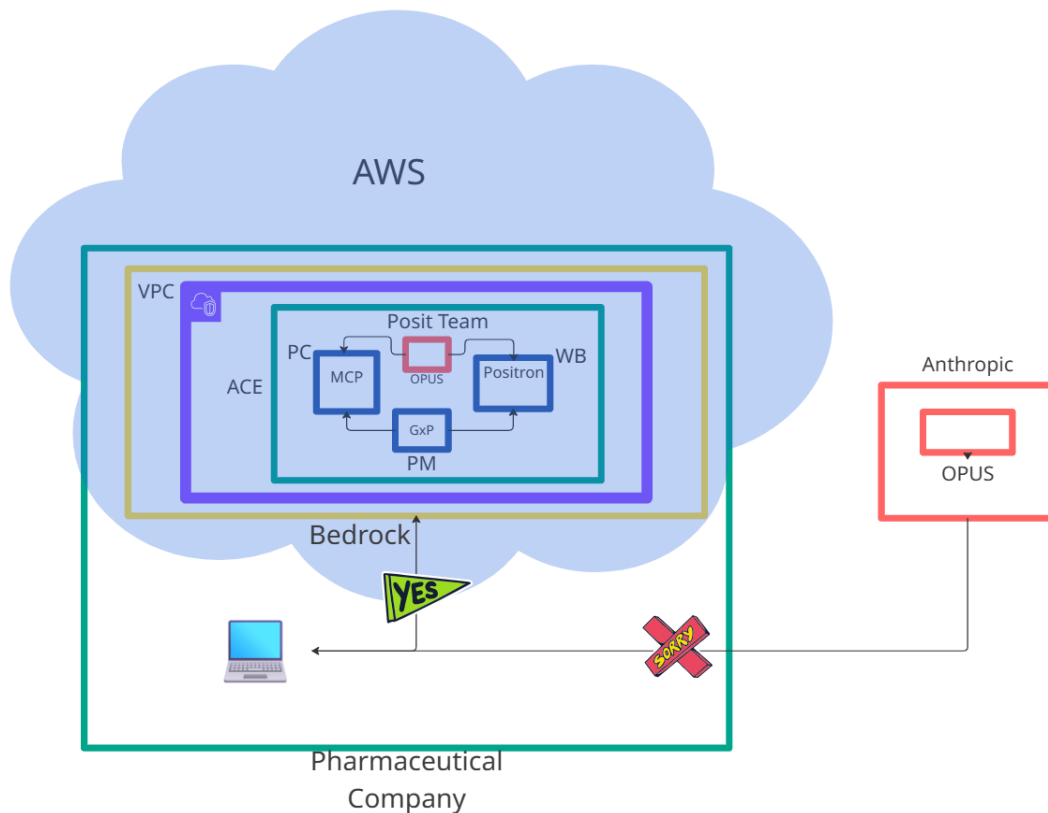
A challenging aspect of using AI in statistical programming is just having access to foundational models within organizations. Curating and fine-tuning local models can be labor intensive and many Pharmaceutical organizations limit access to public foundational models for various reasons such as sensitivity of clinical data. This paper will discuss Amazon Bedrock as a managed service to governed and enterprise-grade access to foundation models. This paper will break down what Bedrock is, how it works and its implications in late-stage reporting. Below will introduce Bedrock and then break it down into use-cases for users and administrators.

## Introduction

At its core, Bedrock is a managed service that provides secure and governed access to popular foundation models from leading AI companies within an organization.

<https://aws.amazon.com/bedrock/>

Bedrock is a service from AWS and runs in the Pharmaceutical's AWS infrastructure. This is usually referred to as a VPC (Virtual Private Cloud). Most Pharmaceutical organizations have a SCE (Statistical Computing Environment) in a VPC. The SCE is evolving into an ACE (AI Coming Environment) where organizations have governed AI available for users and agents. Below is a simple network diagram of how pharmaceutical companies use Bedrock to provide Anthropic Opus:



Because users are in an ACE (AI Computing Environment) within the pharmaceutical's secure AWS VPC, surrounded with the Bedrock fabric, there is a silent handshake where Positron and its R and Python packages connect to the *AWS Instance Metadata Service (IMDS)*. IMDS is the bridge that allows the code to connect by default without requiring special configs or user settings by setting an approved Identity and Access Management (IAM) role. IMDS via AWS allows VPCs to access information about themselves and includes Claude models like Opus. AI requests are sent through the VPC Interface Endpoint which keeps data inside the pharma network and AWS's private fiber. By using *us.* model IDs, there is a guarantee that clinical data remains within the US AWS infrastructure and behind the firewall. Moreover, a Zero Data Retention policy within AWS keeps data in the memory within AWS, keeping the data and request within the pharma and never sent to Anthropic.

## Part 1: Bedrock for Users via Positron

Positron, the new AI-driven IDE from Posit, has AI capabilities built in. There are 3 primary ways to leverage AI in Positron:

1. Posit Assistant
2. Claude Code
3. Open-Source Packages

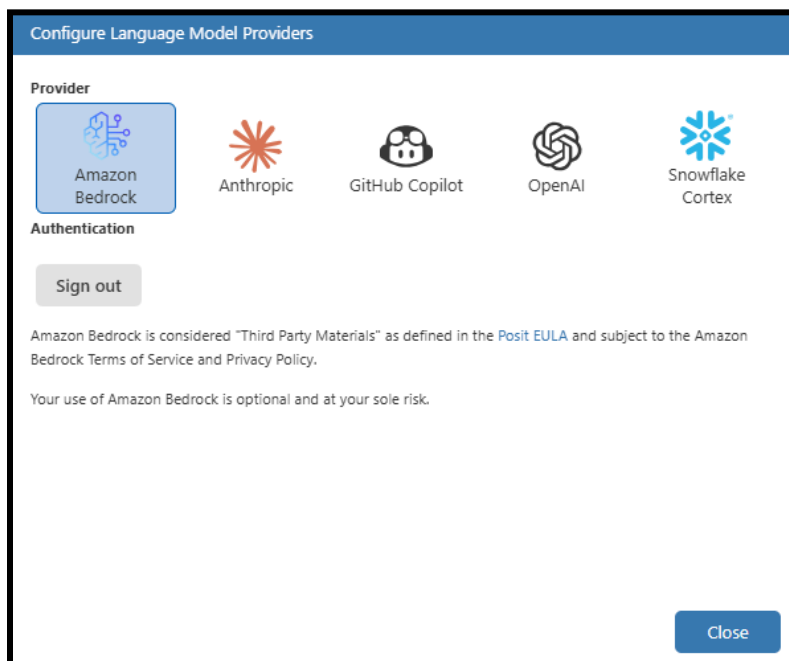
Below we will explain these 2 capabilities as they will be referenced within the paper.

### Posit Assistant

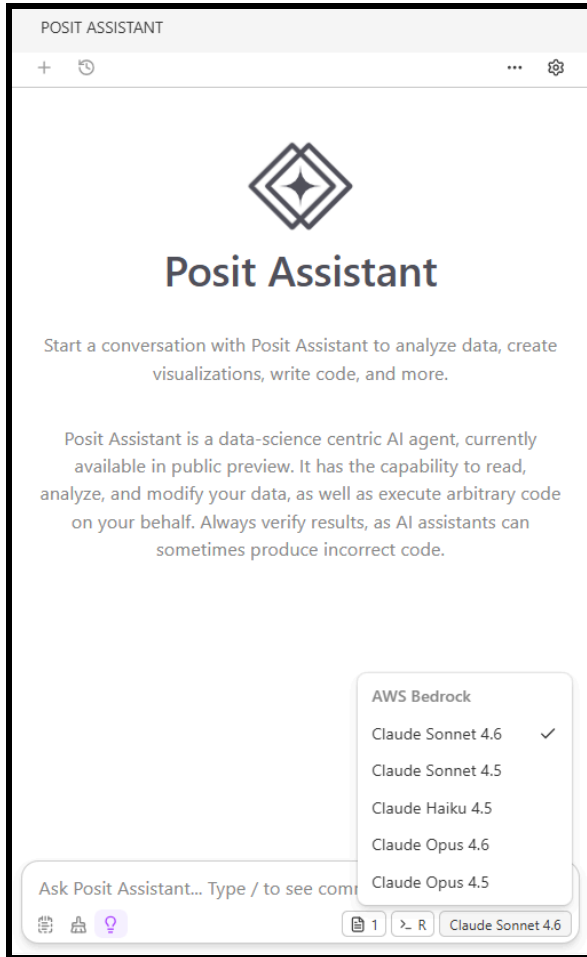
Posit Assistant is an AI coding assistant built specifically for statistical programming workflows.

While many AI tools (Claude Code etc.) are great for software engineering, many are challenged with common statistical programming tasks and suffer out-of-the-box from *Session Blindness*. This makes it difficult for Claude Code and other tools to use in-memory data, know which packages to use for GxP, and show output in the IDE.

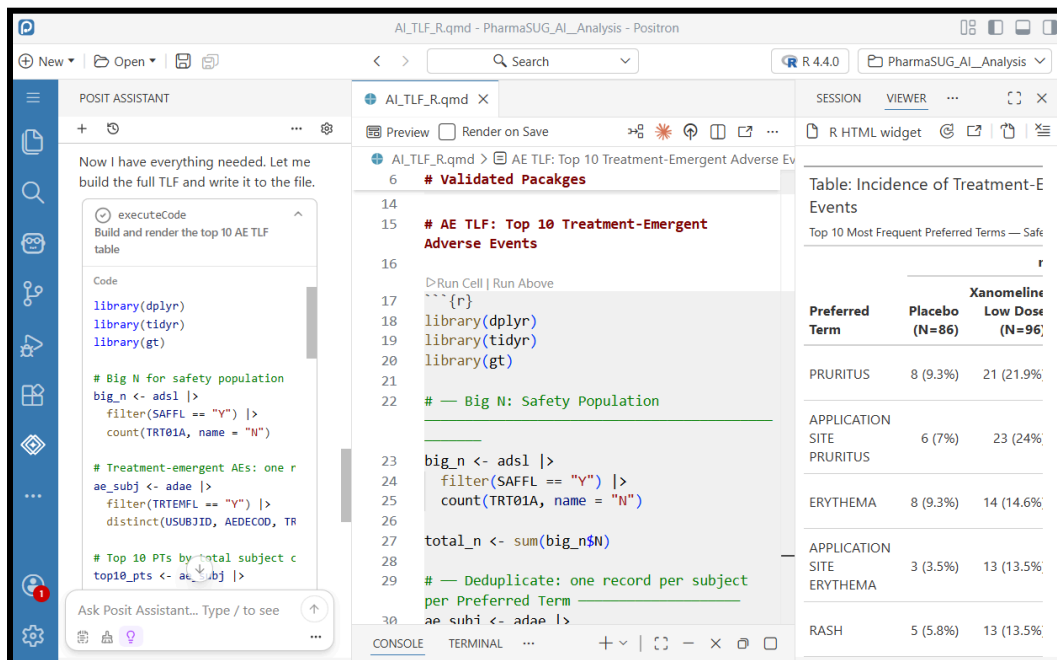
Posit Assistant was built to solve these challenges instantly. Below will show how to use Posit Assistant with Bedrock. Most organizations IT departments will pre-set Bedrock as the default provider:



Then when users launch the Posit Assistant, it will be ready to go with the default Foundational models. Below are the Claude models:



Users now can prompt as needed and use Claude Sonnet via Bedrock as in the TLF example below:

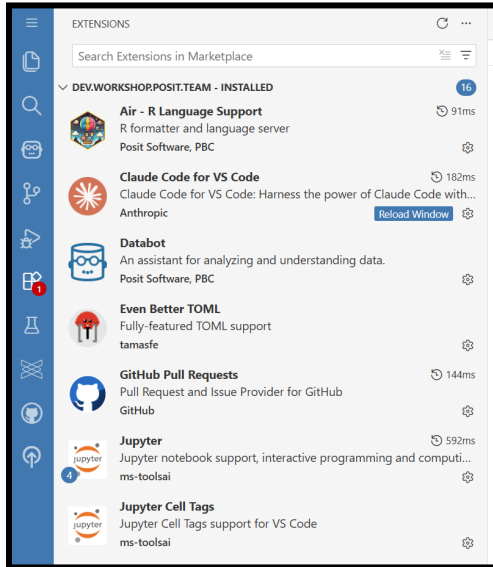


## Claude Code

In addition to using the Posit Assistant, users can use Claude Code. Claude Code can be instructed to use Claude models via Bedrock using the settings below.

## Install the Claude Code Extension

Install the Claude Code OpenVSX Extension: <https://open-vsx.org/extension/Anthropic/claude-code>



## Configure Claude Code to use Bedrock

Below will explain how to use Amazon Bedrock with Claude Code.

Disable the Claude Code login prompt in your Positron user settings:

```
JSON
{
  "claudeCode.disableLoginPrompt": true
}
```

Configure Claude Code to use AWS Bedrock. This example will use single-signon. See <https://code.claude.com/docs/en/vs-code#use-third-party-providers> for more details on configuring AWS Bedrock. Add the following to `~/.claude/settings.json`:

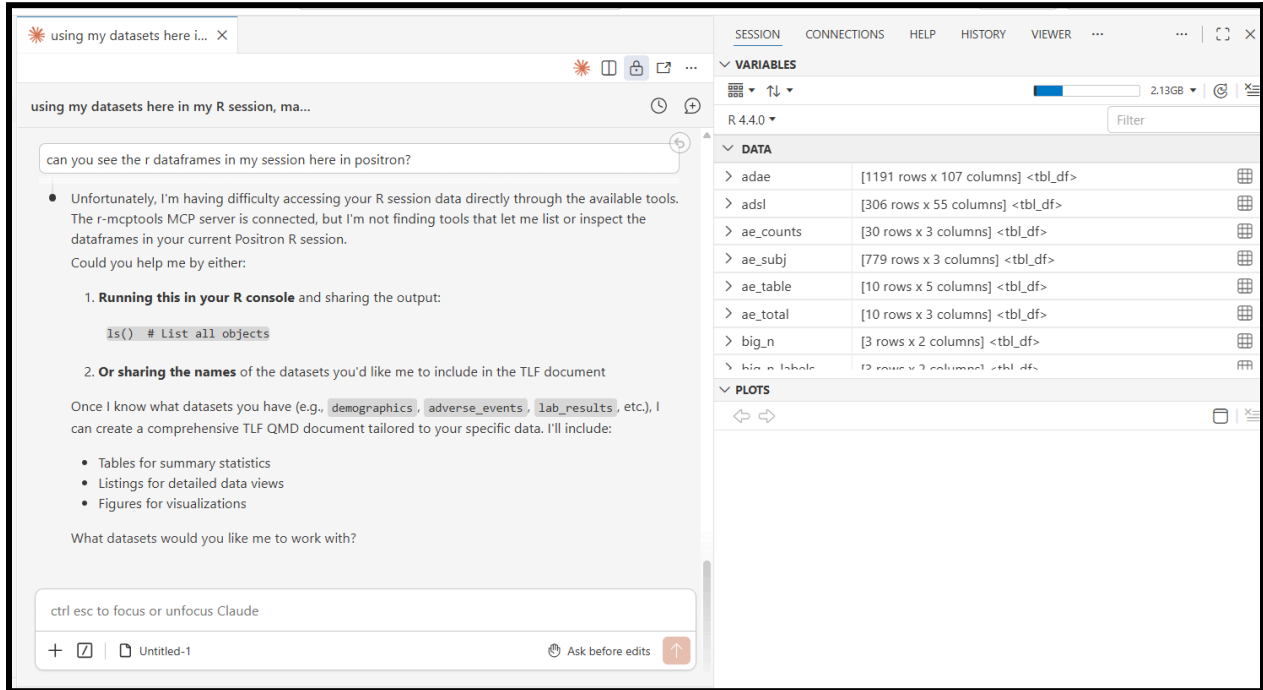
```
JSON
{
  "$schema": "https://json.schemastore.org/claude-code-settings.json",
  "env": {
    "CLAUDE_CODE_USE_BEDROCK": "1"
  }
}
```



## Use Claude Code

Users can open Claude Code by clicking the follow symbol:

This will open the Claude Code extension or users can use Claude via the Terminal by typing Claude. Please note without MCPs configured, Claude Code may have difficulty working with clinical data in Positron.



Please note that

## Open-Source Tools

In addition to using built-in AI tools like Posit Assistant, or an extension like Claude Code, users may want to use direct connections to Bedrock for programmatic development. In R and Python, this is completed via two primary tools:

The primary R package is `ellmer`: <https://github.com/tidyverse/ellmer>

The primary Python package is `chatlas`: <https://posit-dev.github.io/chatlas/>

In R, users can connect to Bedrock with `ellmer` using the `chat_aws_bedrock` function.

None

```
library(ellmer)
```

```
chat <- chat_aws_bedrock(model = 'us.anthropic.claude-sonnet-4-5-20250929-v1:0')
```

```
chat$chat("create a shiny app using library(pharmaverseadam)")
```

In Python, users can connect to Bedrock with `chatlas` using the `ChatBedrockAnthropic` function.

```
Python
import chatlas as ctl
import requests

chat = ctl.ChatBedrockAnthropic(
    model="us.anthropic.claude-sonnet-4-5-20250929-v1:0"
)

chat.chat("create a shiny for python app using openfda data")
```

## Model Providers/Platforms

Posit's AI capabilities work with many different model providers and platforms, such as AWS Bedrock, Azure AI Foundry, and Google Vertex AI. etc. Below will focus on Bedrock.

### AWS Bedrock

AWS Bedrock helps in keeping data private, allows users to pick different models, and are pay as you go (tokens). Given these advancements to empower the ACE, there are still challenges. There are often complex setups where new models require a separate marketplace agreement. Centralized logging across accounts and regions can be a significant engineering hurdle. Financial governance can also be demanding; although new 2026 features allow for tagging by IAM user, tracking granular per-user costs in a rapid growth environment still requires proactive AWS Budget Alerts to prevent runaway spend. Given the rapid release of frontier models like Claude 4.5, traditional anomaly detection is still catching up, making automated alerts and rigorous infrastructure-level controls the only reliable way to maintain fiscal stability.

Positron Pro 2025.10.1 introduced the ability to connect to Bedrock. To enable Positron Assistant with AWS Bedrock, users add the following specific user-level Positron settings in settings.json:

```
JSON
"positron.assistant.provider.amazonBedrock.enable": true
```

If wanted, IT can also force the Posit Assistant to use certain models by default so fall into the right one when launching a session:

```
JSON
"positron.assistant.models.preference.amazonBedrock": "Claude 3.5 Opus"
```

## Conclusion & Contact

The information above highlights the evolving AI ecosystem of AI Computing Environments (ACE) and using Bedrock as an AI fabric that provides a secure method for using sensitive clinical data.

Phil Bowsher

[phil@posit.co](mailto:phil@posit.co)

<https://github.com/philbowsher>