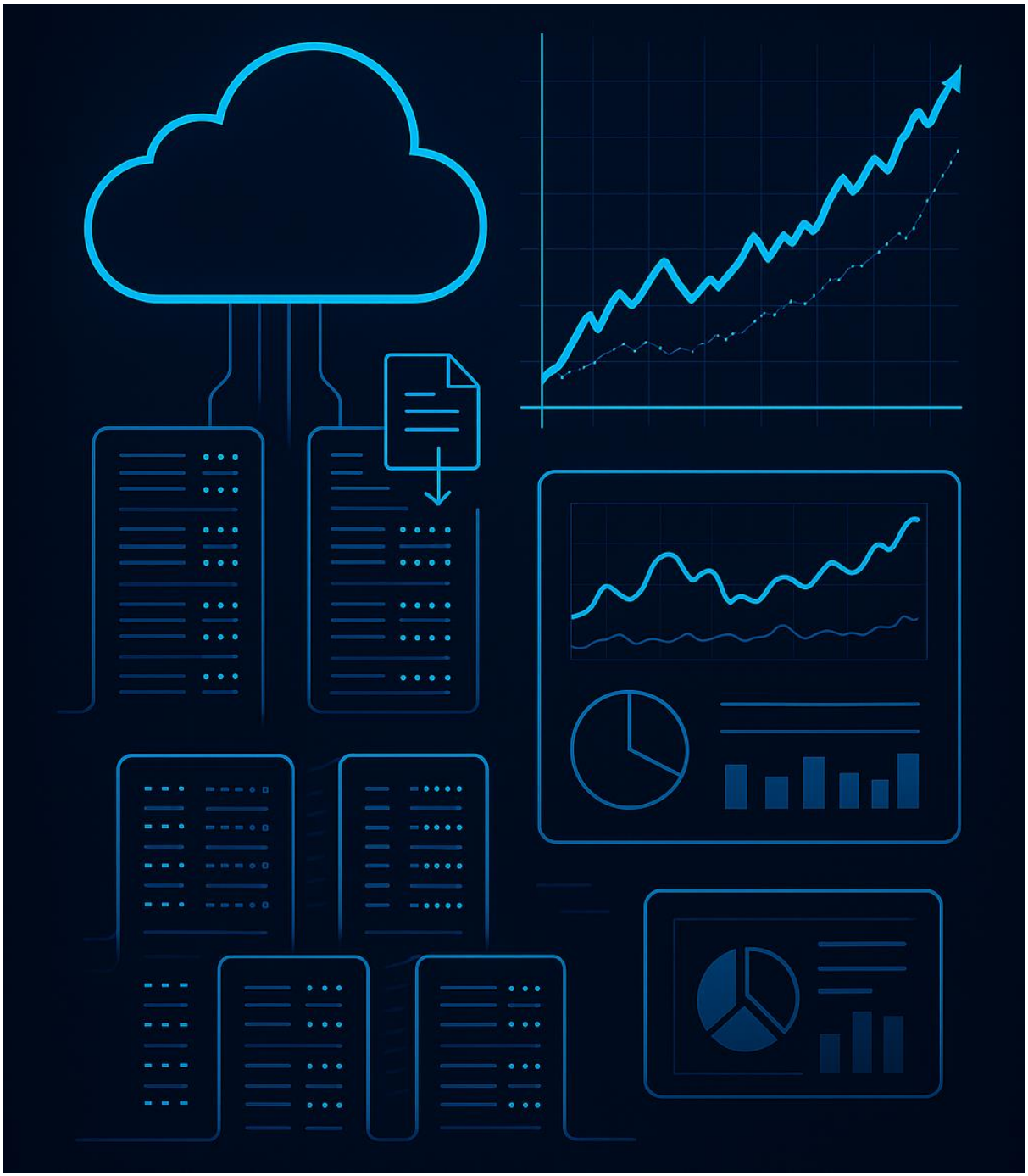


Demand Forecasting and Capacity Planning for Cloud Infrastructure

Team Member: Mohanasundaram Murugesan

NetId: vw4192

Subject: BAN 673 – Spring 2025

Executive Summary

This project focuses on forecasting normalized cloud infrastructure demand using a production-scale dataset published in the research paper *"Shaved Ice: Generating Demand Forecasts from Normalized Cloud Usage Data."* The dataset contains hourly compute usage across multiple geographic regions and virtual machine instance types. To simplify modelling and enhance interpretability, usage data was aggregated into a single hourly time series representing total normalized demand.

Exploratory analysis revealed a strong Daily **weekly seasonal pattern** and a gradual **upward trend**. STL decomposition confirmed these components, validating the use of forecasting models that capture both trend and seasonality.

The project evaluated three forecasting approaches:

1. **Holt-Winters Exponential Smoothing (ETS)**: A state space model automatically selected to capture level, trend, and seasonality.
2. **Two-Level Model (Regression + AR(1))**: A hybrid approach combining a linear trend-and-seasonality regression with an AR(1) model on residuals, effectively addressing autocorrelation.
3. **Auto-ARIMA**: An automatically selected ARIMA model that identifies optimal orders of autoregression, differencing, and moving average, including seasonal components.

Each model was trained on approximately three years of hourly data and tested on an 8-week validation set. Forecast accuracy was assessed using RMSE and MAPE. **Holt-Winters and Auto-ARIMA** produced the most accurate results, while the **two-level model** provided strong interpretability and good performance. ACF plots of residuals confirmed that model assumptions were reasonably met.

Final models were used to generate **two-week-ahead forecasts** to support short-term cloud capacity planning. These forecasts provide a data-driven basis for managing resource commitments, balancing cost-efficiency with operational reliability.

Overall, this project demonstrates the effectiveness of time series forecasting as a strategic tool for infrastructure planning. The approach is scalable and adaptable to more granular forecasting by region or SKU. Future enhancements could include incorporating exogenous variables such as customer activity, pricing dynamics, or promotional campaigns to further improve forecasting accuracy and business value.

Introduction

The rapid growth of cloud computing has transformed how organizations store, process, and deliver digital services. As businesses continue migrating workloads to the cloud, the demand for scalable and cost-efficient infrastructure has surged. Providers like AWS, Azure, and Snowflake face mounting pressure to balance capacity planning with dynamic, often unpredictable usage patterns. Overprovisioning leads to unnecessary costs, while under provisioning risks degraded service performance and lost revenue opportunities.

To navigate this challenge, cloud providers rely on accurate demand forecasts to inform strategic decisions around infrastructure procurement, service availability, and cost optimization. Usage patterns are rarely consistent. Workloads fluctuate due to seasonality, customer behaviour, product launches, and global events. This variability makes robust time series forecasting a critical tool in cloud operations planning.

This study explores historical cloud compute usage data, captured hourly from February 2021 to January 2024, and focuses on forecasting total normalized demand across all regions and virtual machine instance types. The dataset, derived from the research paper *"Shaved Ice: Generating Demand Forecasts from Normalized Cloud Usage Data,"* offers a rich view of cloud resource consumption over time. To streamline analysis, usage across all regions and instance types was consolidated into a single hourly series representing aggregate demand.

By applying statistical forecasting methods such as Holt-Winters Exponential Smoothing (ETS), Auto-ARIMA, and a two-level regression model augmented with AR(1) error correction, this project aims to deliver accurate short- to medium-term forecasts that can support strategic infrastructure decisions such as capacity planning, autoscaling policies, or long-term procurement. The goal is not only to improve predictive accuracy but also to translate technical results into actionable business recommendations for infrastructure provisioning, cost control, and operational efficiency.

Understanding the patterns driving cloud demand and forecasting them reliably is essential for ensuring that cloud services remain resilient, responsive, and cost-effective in a highly competitive digital ecosystem.

Eight Steps of Forecasting

Step 1: Define Goal

This report presents a comprehensive time series forecasting analysis to predict cloud compute resource demand. The study is based on a normalized hourly usage dataset from February 2021 to January 2024, derived from the research publication *"Shaved Ice: Generating Demand Forecasts from Normalized Cloud Usage Data."* The dataset aggregates compute usage across multiple regions and virtual machine instance types, and was consolidated into a single hourly time series to streamline analysis.

We applied a range of statistical forecasting techniques to model and predict future demand, including:

1. **Seasonal Naïve Forecasts**
2. **Holt-Winters Exponential Smoothing (ETS)**
3. **Linear Regression with Trend and Seasonality**
4. **Two-Level Forecasting (Regression + AR(1) on Residuals)**
5. **Auto-ARIMA Model**

Each model was trained on historical data and validated using an 8-week hold-out set. Forecast performance was evaluated using Root Mean Square Error (RMSE) and Mean Absolute Percentage Error (MAPE) to determine the most accurate and robust approach.

The primary objectives of this study are to:

1. Identify long-term trends and daily/weekly seasonality in cloud usage.
2. Compare forecasting models to evaluate accuracy and interpretability.
3. Generate short- to medium-term forecasts to support cloud capacity planning.

Accurately forecasting infrastructure demand is essential for cloud providers to balance cost-efficiency with performance. These forecasts can inform resource commitment decisions, autoscaling strategies, and procurement planning in large-scale multi-cloud environments.

Step 2: Get Data

Dataset Source

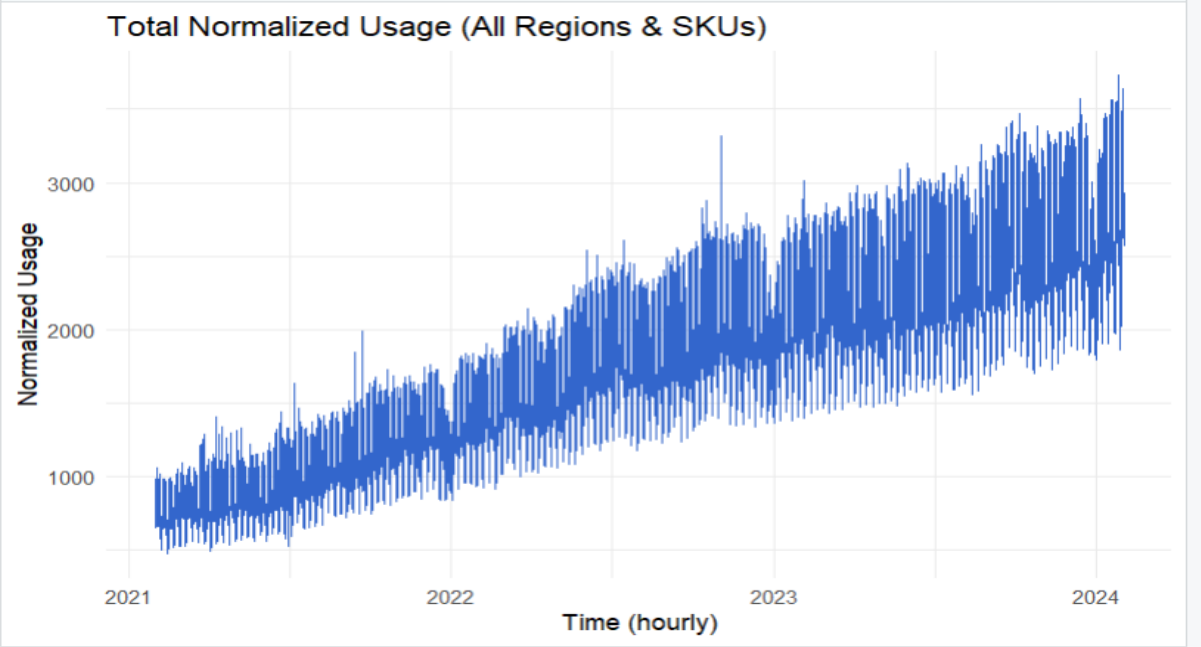
- The dataset is provided by **Snowflake Inc.** and published as part of the academic research paper *"Shaved Ice: Generating Demand Forecasts from Normalized Cloud Usage Data"*.
- It contains **hourly virtual machine (VM) usage data** from **February 2021 to January 2024**, collected across **multiple cloud regions** and **instance types**.
- The dataset was made publicly available to support cloud infrastructure planning research and includes over **26,000 hourly observations**.

Key Data Columns

Column	Description
USAGE_HOUR	Timestamp in hourly intervals (character format, later parsed)
REGION_NUM	Region identifier (e.g., cloud provider zone)
INSTANCE_TYPE	VM instance type (e.g., size or configuration class)
NORM_USAGE	Normalized VM usage (unitless index, scaled across SKUs)

Step 3: Explore and Visualize the Data

Initial Visualization



The plot above shows the total normalized cloud usage aggregated across all regions and virtual machine types from February 2021 to January 2024. Each point represents hourly usage, and the data spans nearly three years.

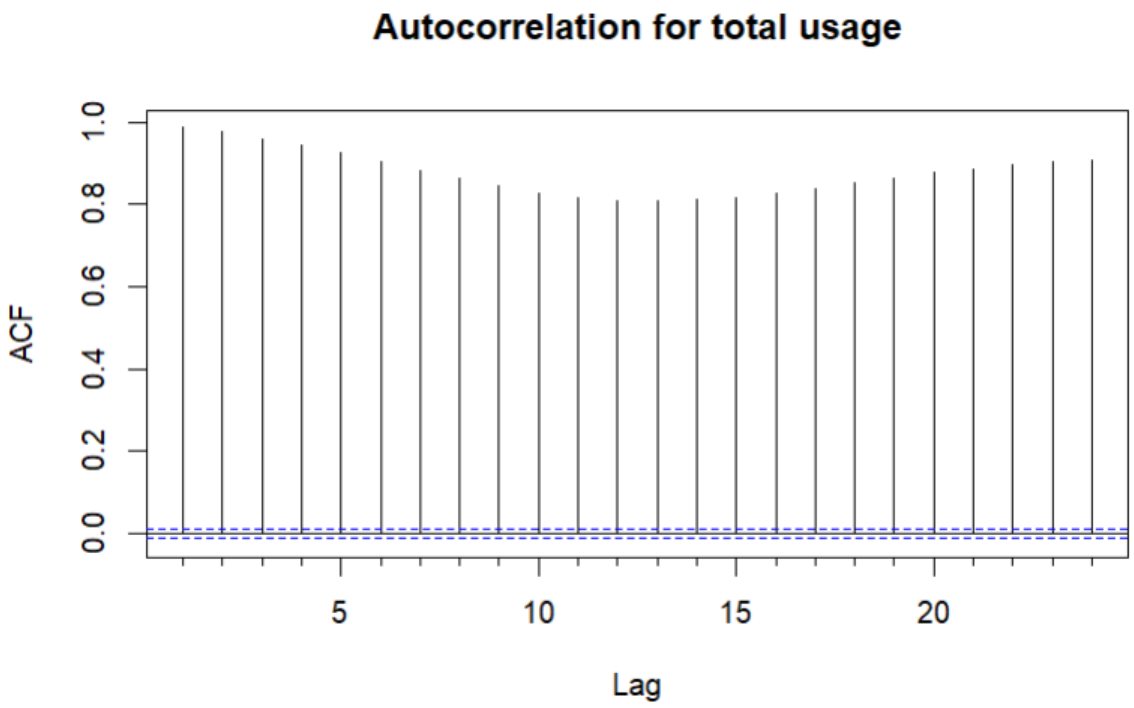
The visualization reveals two key patterns:

1. Clear Upward Trend: Usage steadily increases over time, indicating growing demand for cloud compute resources.
2. Strong Seasonality: The sharp vertical fluctuations within each day reflect recurring daily cycles, while broader wave-like patterns suggest weekly and possibly monthly seasonality.

Auto correlation Function:

The **Autocorrelation Function (ACF)** measures how strongly a time series is correlated with its own past (lagged) values. It is a foundational tool in time series analysis and helps uncover:

- **Persistence** – How long past values influence future values.
- **Seasonality** – Whether patterns repeat over consistent time intervals.
- **Model suitability** – Whether models like **AR(1)**, **MA(1)**, or **ARIMA** are appropriate.



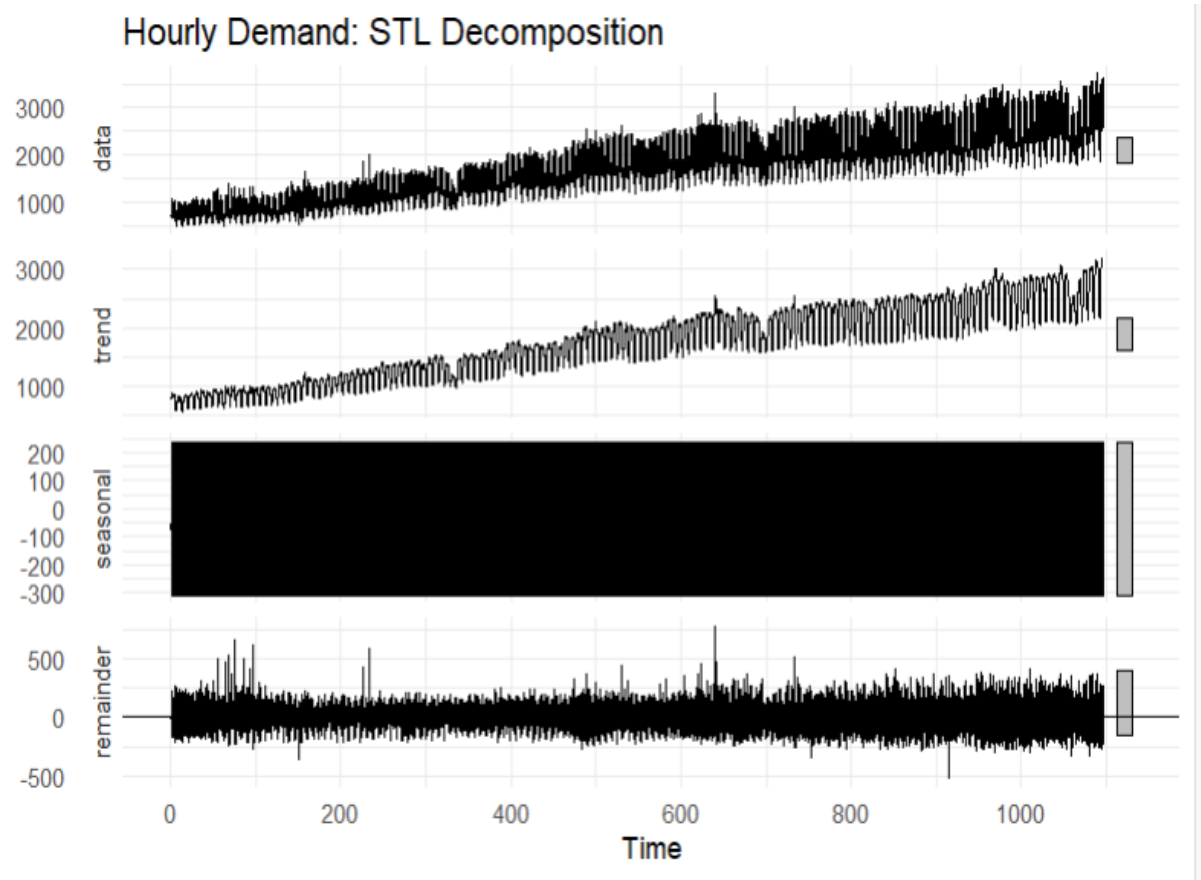
Interpreting the Plot: Autocorrelation for Total Usage

The ACF plot displays autocorrelations of **total hourly cloud usage** at lags 1 through 24.

Key Observations:

Feature	Interpretation
Very high autocorrelations (≈ 1)	The series is **highly persistent**. Current values are strongly influenced by recent past values.
Slow decay across lags	Suggests a **non-stationary trend** possibly due to long-term growth in cloud usage.
No dip at lag 24	Unexpected for hourly data with daily cycles. This may be **masked by trend**, or daily seasonality is less dominant and needs **differencing** or **STL decomposition** to be clearly visible.

STL Decomposition: Hourly Cloud Usage (Daily Seasonality):



Data (Top Panel) – Raw Hourly Usage

- Represents the original time series, aggregated on an hourly basis.
- Key patterns:
    - Clear upward trend over time, indicating growing compute demand.
    - High-frequency fluctuations that align with daily usage cycles.
    - Occasional periods of volatility, suggesting dynamic workload changes.

Trend – Long-Term Pattern

- Extracts the underlying growth trend by smoothing out short-term variation.
- Key observations:
    - Steady and gradual increase in usage from 2021 to 2024.
    - Subtle inflection points may reflect seasonal shifts, infrastructure upgrades, or organizational scaling patterns.

Seasonal – Daily Cycle (24-Hour Pattern)

- Captures the repeating daily pattern present in the series due to regular usage cycles.
- Visual note:
    - The panel appears dense or black because STL estimates 24 seasonal values per day across roughly 1,000+ days, resulting in extreme overplotting.
- Interpretation:
    - Daily seasonality is clearly present, but the plot is too dense to interpret visually. Zooming into a smaller time window may help reveal the pattern more clearly.

Remainder – Irregular or Unexplained Variation

- Represents residuals left after removing both trend and seasonal components.
- Key takeaways:
    - Residuals are centered around zero, which suggests a well-fitted model.
    - A few isolated spikes may represent outliers or sudden usage surges.
    - Overall, the residuals appear stable, indicating the decomposition effectively captured the structure in the data.

Hypothesis Testing for Predictability

- An **AR(1) model** was fitted, and a **Z-test** was performed on the autoregressive coefficient.
- **Result:** The **null hypothesis was rejected**, confirming that the dataset exhibits **predictable patterns**, making it suitable for time series forecasting.

Step 4: Data Preprocessing

1. Clean and Parse Timestamps

To ensure accurate time-based operations, the USAGE_HOUR column was cleaned and parsed into proper datetime format (POSIXct). This included:

- Removing extra spaces between date and time (e.g., "01-02-2021 00:00:00" → "01-02-2021 00:00:00").
- Stripping millisecond suffixes (e.g., ".000"), which can interfere with datetime parsing.
- Parsing flexible datetime formats using the parse_date_time() function, which handled both:
    - Day-Month-Year (dmy)
    - Year-Month-Day (ymd)
    - With or without seconds

This step ensured uniform and consistent hourly timestamps across the dataset.

2. Validate Timestamp Parsing

After parsing, a check was performed to detect any rows with unparsed (NA) timestamps:
num_bad <- sum(is.na(cloud.data$USAGE_HOUR))

If any malformed timestamps remained, the process would halt with an error. In this case, all timestamps were successfully parsed.

3. Aggregate Usage to Hourly Totals

The raw dataset contains **multiple rows per hour**, segmented by region and instance type. To simplify the modeling process:

- All NORM_USAGE values were **aggregated (summed)** across each hour using:

  group_by(USAGE_HOUR) %>% summarise(total_usage = sum(...))

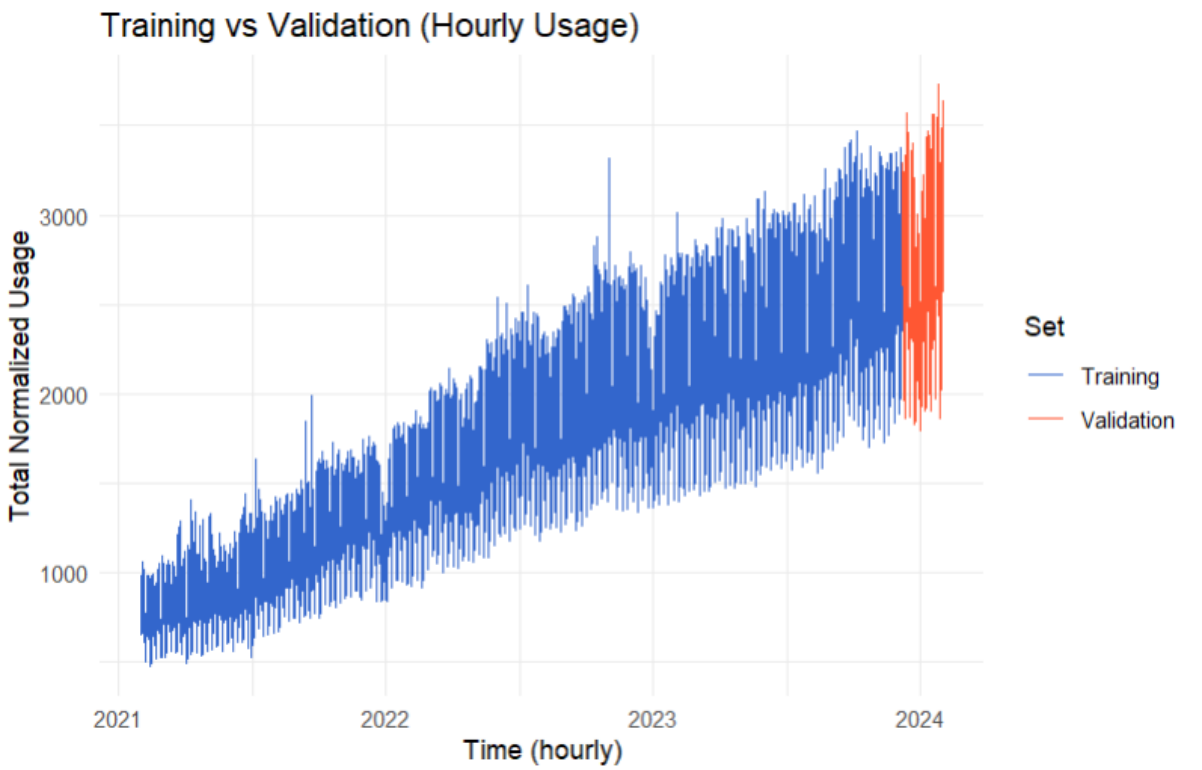
- This resulted in a **single aggregated time series** representing **total normalized cloud demand per hour**, which is well-suited for univariate time series forecasting models.

The final data.all dataset is a clean, ordered, hourly time series with two columns:

- USAGE_HOUR: The timestamp (POSIXct)
- total_usage: The summed demand value for that hour

Step 5: Partition the Series

Splitting into Training & Validation Sets



- **Training Data**:
  Hourly usage data from **February 2021 to early December 2023**, comprising approximately **95% of the full dataset**. This portion was used to fit all forecasting models.
- **Validation Data**:
  The **last 8 weeks** of the dataset, covering **1,344 hourly observations**, from **mid-December 2023 to January 2024**. This segment was withheld from model training to assess forecast accuracy on unseen data.

Why Use an 8-Week Validation Window?

- **Relevant Horizon**:
  In real-world cloud operations, **short- to medium-term forecasting (1–8 weeks)** is a common planning horizon for resource procurement and scaling decisions.

- **Sufficient Data Size**:
  An 8-week holdout equates to **1,344 hourly observations**, which is large enough to validate the model's ability to generalize, while preserving the majority of the dataset for training.
- **Balanced Evaluation**:
  The chosen window provides a realistic test of each model's ability to handle **both trend and seasonality**, especially for models like ETS and Auto-ARIMA.
- **Consistent Comparison**:
  Using a fixed holdout period across all models allows for fair and consistent comparison of **forecast accuracy** using metrics like RMSE and MAPE.

Note: While longer validation windows can provide additional robustness, the 8-week window strikes a practical balance between evaluation depth and model training power for high-frequency (hourly) data.

Step 6 & 7: Apply and Forecast

Holt's and Winter's Model

The Holt-Winters Model is chosen to forecast our dataset with both seasonality and trend because the Holt-Winters Model effectively captures long-term upward or downward trends through its trend component, which is useful for time series with a clear directional change. The seasonal component in the Holt-Winters model can also efficiently model cyclic patterns.

Besides, there is another significant advantage: we can automatically select the error, trend, and seasonality components for the Holt-Winters model, which reduces model selection complexity, improves model performance and saves users' time.

```
> summary(hw_model)
ETS(M,Ad,N)

Call:
ets(y = train.ts, model = "ZZZ")

  Smoothing parameters:
    alpha = 0.9037
    beta  = 0.1411
    phi   = 0.8

  Initial states:
    l = 637.8157
    b = 33.0288

  sigma:  0.0591

     AIC     AICc      BIC
478441.4 478441.4 478490.1

Training set error measures:
                    ME     RMSE      MAE         MPE     MAPE      MASE        ACF1
Training set 0.05393473 97.18557 71.7041 -0.0899694 4.350004 0.9824838 0.03011575
>
```

ETS Model Summary and Interpretation

This model is an Exponential Smoothing State Space Model (ETS) automatically selected by R using the "ZZZ" option, which tests various combinations of error, trend, and seasonality components.

The selected model is:

ETS(M, Ad, N)

- M (Multiplicative errors): Forecast errors scale with the level of the time series.
- Ad (Additive damped trend): The model includes a linear trend component that gradually levels off (damped).
- N (No seasonality): No explicit seasonal component is included in the model.

This configuration suggests that the model captures scaling variation, a non-linear growth trend, and that seasonality is handled implicitly or considered negligible for the chosen forecast horizon.

Smoothing Parameters

- α (Alpha) = 0.9037
  Indicates that the model places strong weight on the most recent observations when updating the level. This is consistent with rapidly adapting to recent shifts in demand.
- β (Beta) = 0.1411
  Controls the contribution of the trend. A moderate value suggests the trend is responsive but not volatile.
- ϕ (Phi) = 0.8
  The damping parameter. Since it's less than 1, it means the trend is expected to flatten out over time rather than continue increasing indefinitely.

Initial States

- **ℓ (Level) = 637.8157**
  The model’s initial estimate of the base hourly demand level.
- **b (Trend) = 33.0288**
  The starting value for the trend component suggesting usage was growing by about **33 units per hour** at the beginning of the training period.

Model Fit Statistics:

Metric	Value	Interpretation
RMSE	97.19	Average forecast error magnitude (in usage units)
MAPE	4.35%	Forecast error as a percent of actual demand a low and acceptable value
ME	0.0539	Mean error (close to 0, indicating no major bias)
MAE	71.70	Mean absolute error (error in actual usage units)
ACF1	0.03	Very low autocorrelation in residuals → model fit is strong

- AIC / BIC / AICc all fall around 478,000, useful for comparing to other models like ARIMA or TBATS.
- The residual standard deviation (σ) is 0.0591, indicating very little unexplained variation after fitting.

Overall Assessment

The ETS(M, Ad, N) model performs well on the training set, capturing both the multiplicative error structure and the damped trend in the cloud usage time series. The model produces a low MAPE (4.35%), and minimal residual autocorrelation, confirming that the model effectively explains the patterns in the training data.

While the model excludes explicit seasonality, this is reasonable given the use of daily frequency (24) and may be supplemented by other modeling techniques (e.g., ARIMA) for multi-season forecasts.

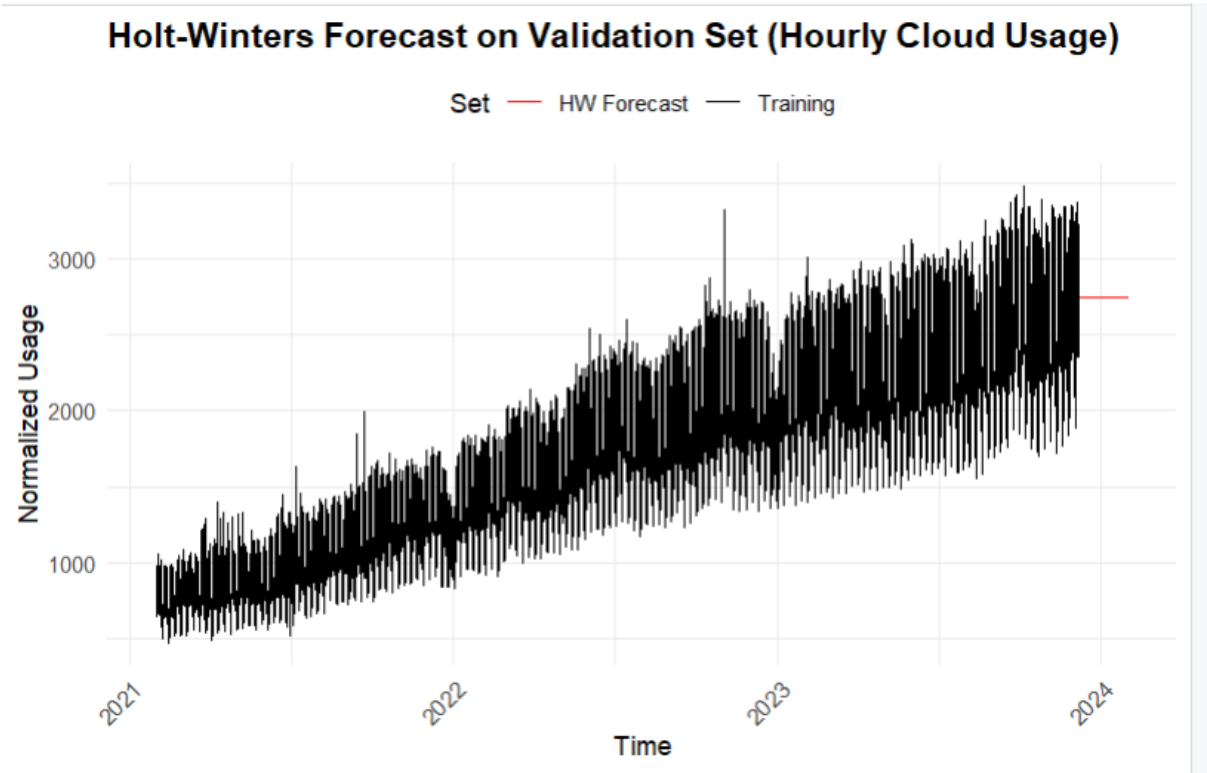
Validation Accuracy – Holt-Winters (ETS) Model

The Holt-Winters model achieved the following performance on the 8-week validation set:

- RMSE: 457.96
- MAE: 394.21

- MAPE: 15.94%
- ME: –97.45 (slight underforecasting bias)

These results indicate moderate forecasting error, with a mean absolute percentage error under 16%, which is acceptable for operational planning. The model slightly underpredicts on average, but captures trend and level reasonably well.



The visualization above shows the performance of Holt-Winters model on the validation partition of the dataset compared with its actual data on the corresponding period. However, according to the red line, the forecast given by Holt-Winters model does not show any visible seasonality and does not fit well with the actual data of the validation partition of the dataset and there is an obvious underestimation on validation partition.

ETS Model Summary (Full Data)

The automatically selected model is ETS(M, Ad, M):

- Multiplicative errors – forecast uncertainty scales with usage levels.
- Additive damped trend – demand is increasing but at a gradually slowing rate.
- Multiplicative seasonality – daily usage cycles vary proportionally with the level.

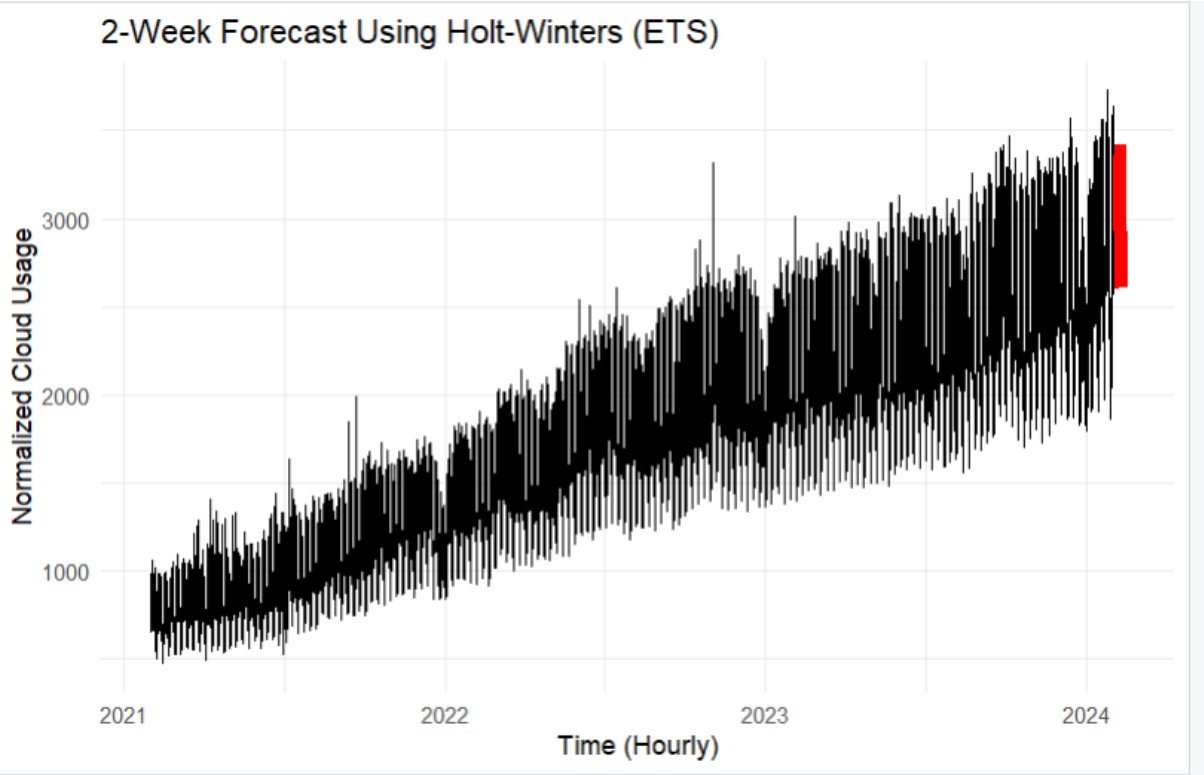
Model Parameters

- Alpha (level): 0.7586 → strong influence from recent data.
- Beta (trend): 0.002 → trend evolves slowly, suggesting stability.
- Gamma (seasonality): 0.0937 → moderate seasonality effect.
- Phi (damping): 0.9342 → trend flattens over time.

Model Fit Metrics

- RMSE: 66.99
- MAE: 50.24
- MAPE: 3.04%
- ACF1 (residual autocorrelation): ~0 → residuals appear uncorrelated

These indicate excellent model fit, with very low forecast error and well-behaved residuals.



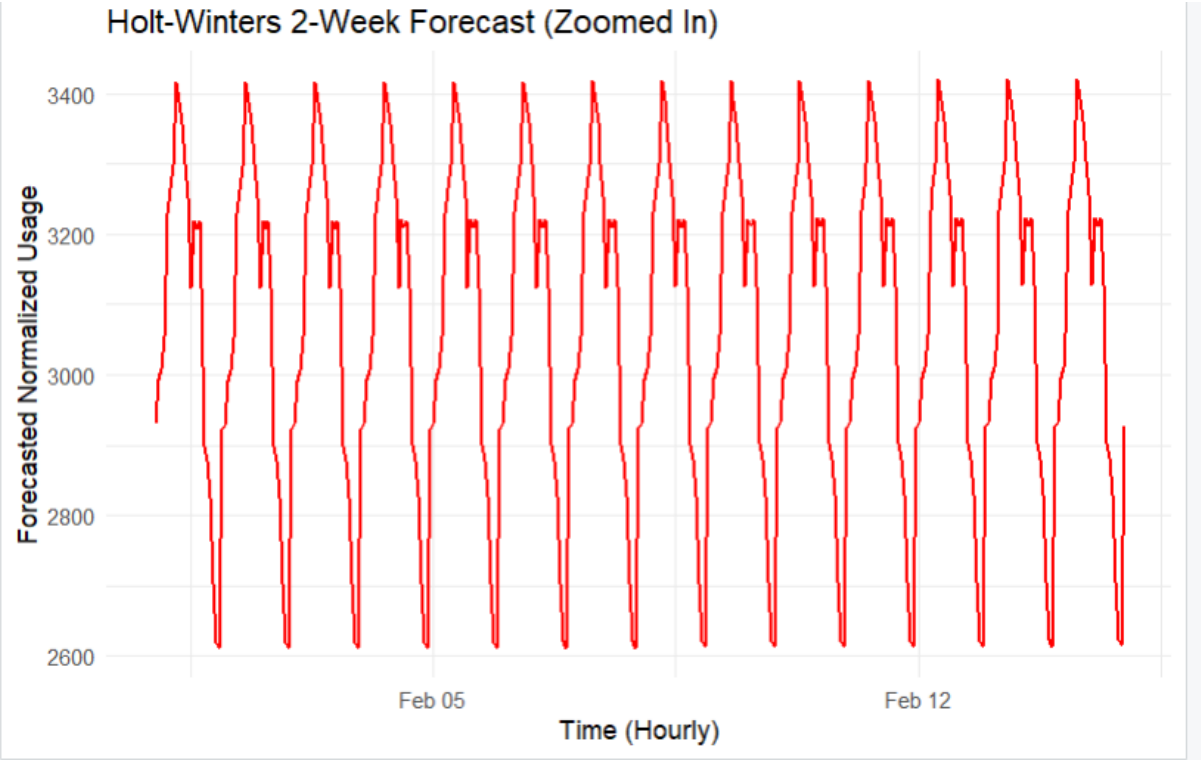


Figure: Holt-Winters 2-Week Forecast (Zoomed In)

The plot shows the forecasted hourly cloud usage for a 2-week horizon using the Holt-Winters exponential smoothing model. The forecast captures:

- Strong daily cycles, with sharp peaks and valleys that repeat every 24 hours.
- A consistent pattern, indicating stable short-term demand fluctuations.
- The amplitude of fluctuations remains relatively stable, reflecting multiplicative seasonality in the ETS model.

This forecast is suitable for short-term operational planning, such as autoscaling or resource reservation, and provides a high-resolution view of expected hourly demand.

Regression Model

We selected Regression Model with Linear Trend + Seasonality because the model includes both trend and seasonality, which matches the nature of our dataset. Hence, we can better forecast the future cloud demand.

```
Call:
tslm(formula = train.ts ~ trend + season)

Residuals:
    Min      1Q  Median      3Q     Max
-852.01 -154.07   52.87  177.20 1353.68

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept)  6.533e+02  8.673e+00  75.320  < 2e-16 ***
trend        7.753e-02  2.319e-04 334.333  < 2e-16 ***
season2      1.881e+01  1.157e+01   1.626  0.10388
season3      3.145e+01  1.157e+01   2.719  0.00655 **
season4      7.688e+01  1.157e+01   6.647 3.05e-11 ***
season5      1.367e+02  1.157e+01  11.816  < 2e-16 ***
season6      2.403e+02  1.157e+01  20.779  < 2e-16 ***
season7      2.726e+02  1.157e+01  23.570  < 2e-16 ***
season8      2.989e+02  1.157e+01  25.842  < 2e-16 ***
season9      3.161e+02  1.157e+01  27.332  < 2e-16 ***
season10     2.814e+02  1.157e+01  24.331  < 2e-16 ***
season11     2.266e+02  1.157e+01  19.589  < 2e-16 ***
season12     2.077e+02  1.157e+01  17.959  < 2e-16 ***
season13     2.021e+02  1.157e+01  17.478  < 2e-16 ***
season14     1.910e+02  1.157e+01  16.518  < 2e-16 ***
season15     1.765e+02  1.157e+01  15.261  < 2e-16 ***
season16     1.238e+02  1.157e+01  10.703  < 2e-16 ***
season17     4.909e+01  1.157e+01   4.244 2.20e-05 ***
season18    -5.920e+01  1.157e+01  -5.119 3.10e-07 ***
season19    -9.282e+01  1.157e+01  -8.025 1.06e-15 ***
season20    -1.036e+02  1.157e+01  -8.955  < 2e-16 ***
season21    -1.525e+02  1.157e+01 -13.184  < 2e-16 ***
season22    -1.933e+02  1.157e+01 -16.710  < 2e-16 ***
season23    -2.216e+02  1.157e+01 -19.162  < 2e-16 ***
season24    -6.684e+01  1.157e+01  -5.780 7.57e-09 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 263.6 on 24911 degrees of freedom
Multiple R-squared:  0.8294,    Adjusted R-squared:  0.8292
F-statistic:  5045 on 24 and 24911 DF,  p-value: < 2.2e-16
```

1. Predictor Variables in the Model

The regression model includes the following predictors:

- Intercept: The baseline level of usage when all other variables are zero.
- trend: A numeric variable that increases linearly over time, capturing the long-term upward growth in cloud usage.
- season2 to season24: These are dummy variables representing hourly seasonality, since you set frequency = 24.

Because there are 24 hours in a day and we're modeling seasonal effects, R creates 23 dummy variables for hours 2 to 24. Hour 1 is absorbed into the intercept to avoid multicollinearity (the dummy variable trap).

2. Understanding Dummy Variables for Seasonality

- season2, season3, ..., season24 represent the effect of each hour of the day on usage, relative to season1 (baseline hour).
- For example:
    - If season7 has a high positive coefficient, that means usage tends to spike during hour 7 (e.g., 6–7 AM) compared to hour 1.
    - If season20 has a negative coefficient, usage during that hour is typically lower than the baseline.

These dummies allow the model to capture **recurring daily fluctuations** in cloud demand.

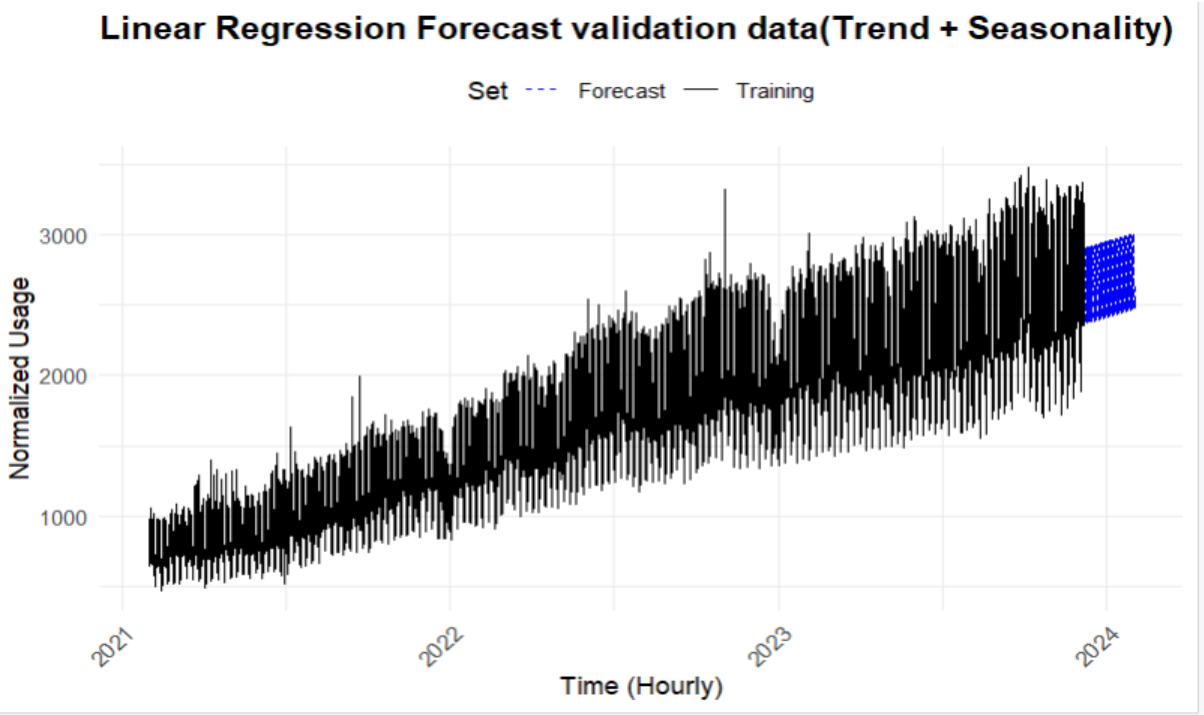
3. R-squared and Adjusted R-squared

- **Multiple R-squared: 0.8294**
  → About **82.94% of the variance** in hourly usage is explained by the trend and seasonal predictors.
- **Adjusted R-squared: 0.8292**
  → Slightly penalized for the number of predictors, but still **very strong**, indicating that your predictors are meaningful and not overfitting.

A value this high confirms that your model captures the underlying structure well, especially given the large sample size (~25,000 hourly records).

4. p-values and Significance

- **p-values < 0.05** (and especially < 0.001) indicate that the coefficient is statistically significant i.e., it likely has a true impact.
- In output:
    - All trend and most seasonX coefficients are **highly significant (p < 2e-16)**.
    - A few, like season2, are marginal (p > 0.05), suggesting that hour may not differ meaningfully from the baseline.

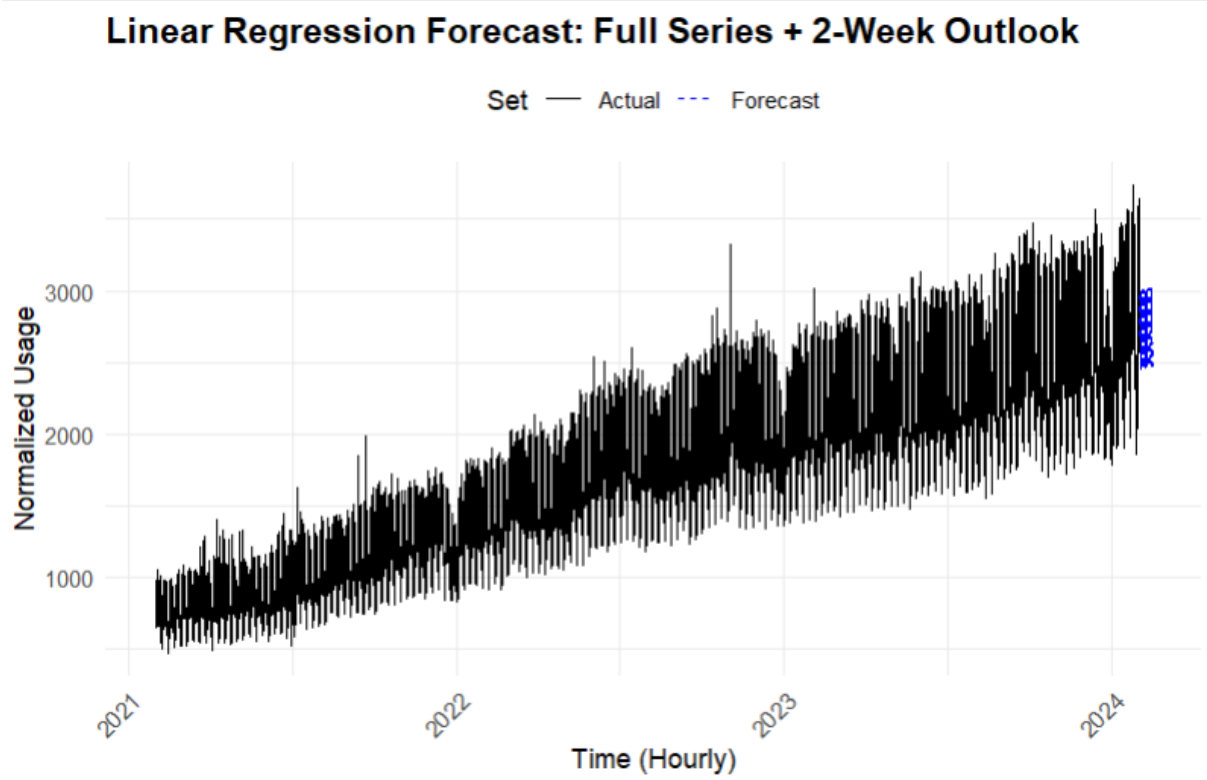


Above plot shows linear regression (trend + seasonality) model forecast on validation data.

```
Call:
tslm(formula = usage.ts ~ trend + season)

Residuals:
    Min      1Q  Median      3Q     Max
-859.78 -163.02   54.51  183.42 1353.31

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept)  6.585e+02  8.717e+00  75.544  < 2e-16 ***
trend        7.674e-02  2.211e-04 346.987  < 2e-16 ***
season2      1.909e+01  1.162e+01   1.642  0.10060
season3      3.066e+01  1.162e+01   2.637  0.00836 **
season4      7.776e+01  1.162e+01   6.690 2.28e-11 ***
season5      1.406e+02  1.162e+01  12.097  < 2e-16 ***
season6      2.437e+02  1.162e+01  20.964  < 2e-16 ***
season7      2.750e+02  1.162e+01  23.662  < 2e-16 ***
season8      3.055e+02  1.162e+01  26.284  < 2e-16 ***
season9      3.196e+02  1.162e+01  27.498  < 2e-16 ***
season10     2.882e+02  1.162e+01  24.797  < 2e-16 ***
season11     2.321e+02  1.162e+01  19.970  < 2e-16 ***
season12     2.112e+02  1.162e+01  18.170  < 2e-16 ***
season13     2.002e+02  1.162e+01  17.221  < 2e-16 ***
season14     1.925e+02  1.162e+01  16.562  < 2e-16 ***
season15     1.785e+02  1.162e+01  15.355  < 2e-16 ***
season16     1.311e+02  1.162e+01  11.282  < 2e-16 ***
season17     5.369e+01  1.162e+01   4.619 3.87e-06 ***
season18    -5.885e+01  1.162e+01  -5.063 4.15e-07 ***
season19    -9.332e+01  1.162e+01  -8.029 1.03e-15 ***
season20    -1.076e+02  1.162e+01  -9.260  < 2e-16 ***
season21    -1.569e+02  1.162e+01 -13.496  < 2e-16 ***
season22    -1.978e+02  1.162e+01 -17.020  < 2e-16 ***
season23    -2.263e+02  1.162e+01 -19.472  < 2e-16 ***
season24    -6.373e+01  1.162e+01  -5.483 4.23e-08 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 272 on 26255 degrees of freedom
Multiple R-squared:  0.8319,    Adjusted R-squared:  0.8317
F-statistic:  5413 on 24 and 26255 DF,  p-value: < 2.2e-16
```

Model Fit Summary – Linear Regression (Trend + Seasonality) full data

- **Adjusted R-squared: 0.8317**
  This means approximately **83.17% of the variation** in normalized hourly cloud usage is explained by the linear trend and seasonal (hour-of-day) components. This reflects a **very strong model fit**, especially for time series data.
- **p-value: < 2.2e-16**
  The model's overall **F-statistic is highly significant**, indicating that the combination of predictors (**trend and seasonality**) **significantly explains** the variation in the data. This confirms that the model is statistically meaningful.



Above plot shows linear regression (trend + seasonality) model forecast on Future (2 weeks)

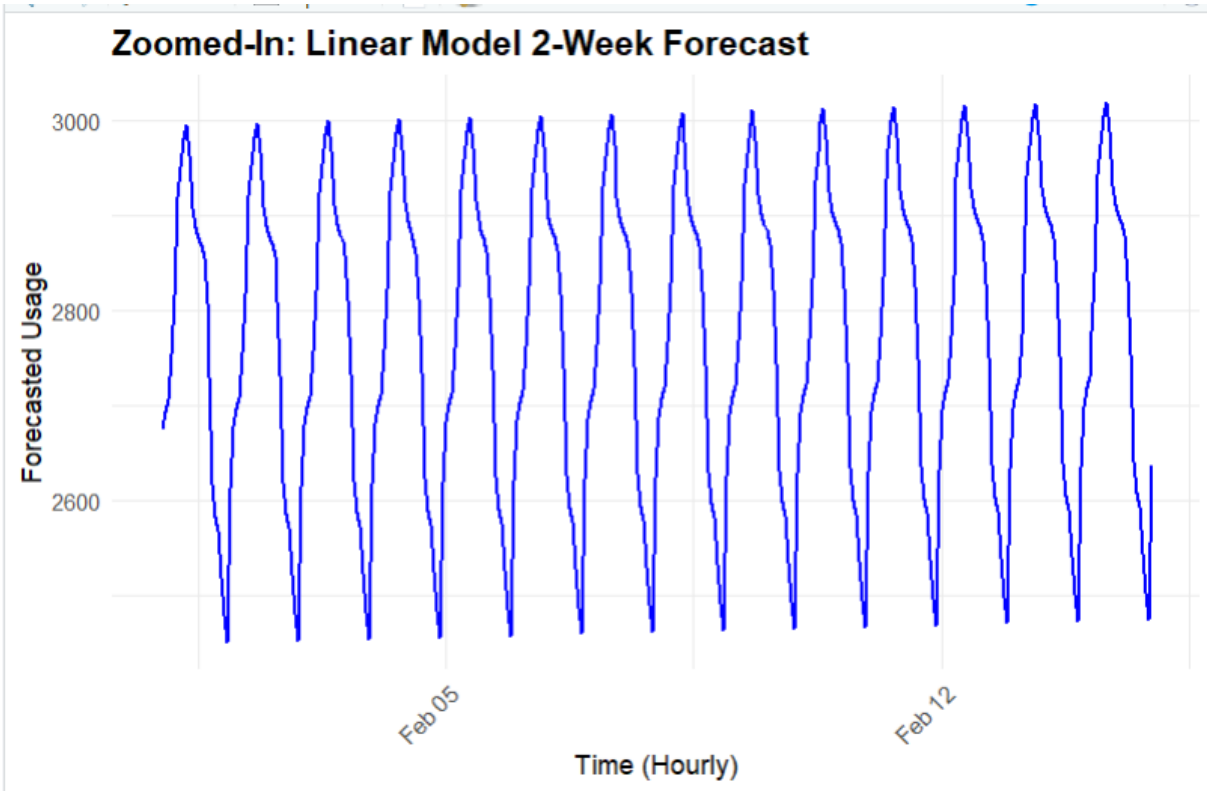


Figure: Zoomed-In Linear Model 2-Week Forecast

This plot presents the **hourly cloud usage forecast** over a 2-week period, generated using a **linear regression model with trend and seasonal components**.

Key observations:

- The model captures a **strong and repeating daily pattern**, with peaks and troughs aligned to typical usage cycles.
- Forecasted usage remains **stable across days**, indicating that the model assumes consistent demand patterns day to day.
- The predicted values reflect the model's **linear trend + fixed hourly seasonal effects**, without adapting to anomalies or external events.
- No significant upward or downward drift is observed over the 2-week horizon, which is expected for a **linear model without evolving seasonality**.

This forecast is suitable for **short-term operational planning**, especially when trends and daily cycles are stable.

<u>Two-Level Model (Regression + AR(1))</u>

To enhance forecast accuracy, we implemented a two-level modeling approach combining a **linear regression model** with an **AR(1) correction on residuals**. The first level captures the overall **trend and daily seasonality** using a regression with time and seasonal dummy variables. While this model explains most of the variation, residual

analysis showed signs of autocorrelation. To address this, an **AR(1) model was fitted to the regression residuals**, allowing the final forecast to account for short-term dependencies. The combined forecast adds the AR(1) predictions to the regression output, improving performance over using regression alone.

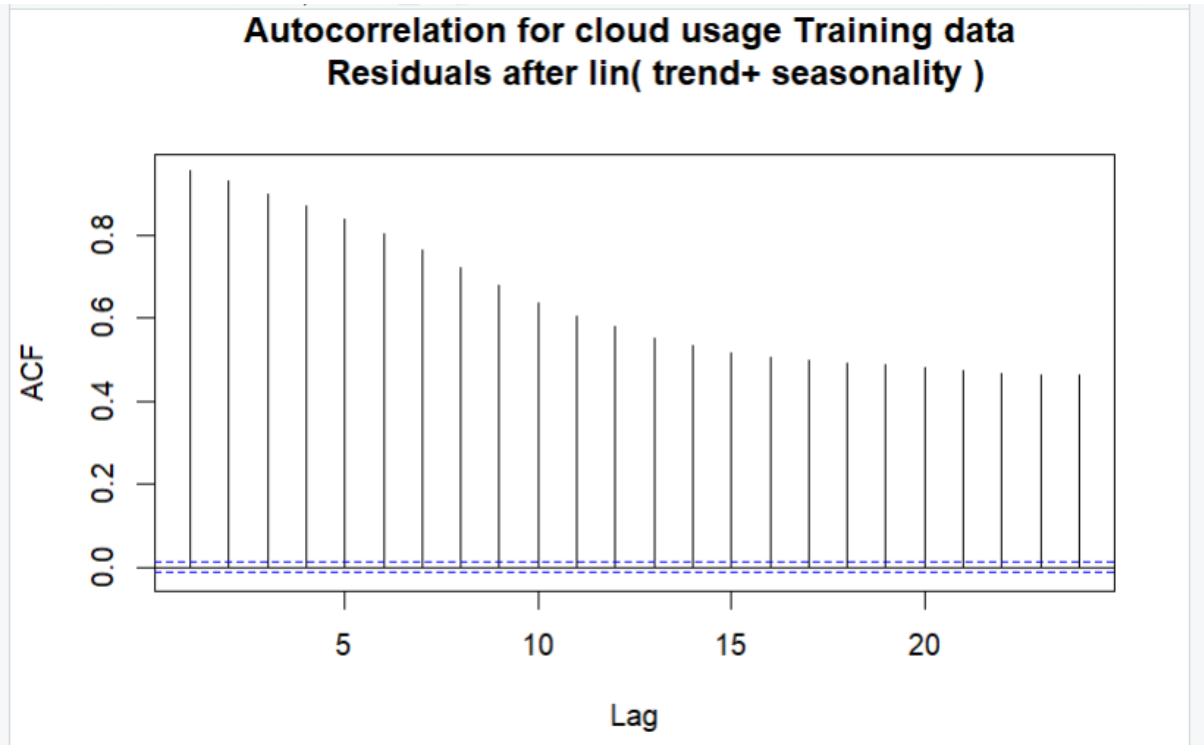


Figure: Autocorrelation of Residuals After Linear Trend + Seasonality Model

This plot shows the **autocorrelation function (ACF)** of the **residuals** from the linear regression model fitted with trend and seasonality on the training data.

Key Observations:

- **High autocorrelation at lag 1 (~0.9)** and a **gradually decaying pattern** suggest that the residuals are **not white noise**.
- The presence of persistent autocorrelation indicates that the linear model, while capturing the trend and seasonality, fails to account for **short-term dependencies** between observations.
- This pattern is characteristic of an **AR(1) process**, where each residual is strongly correlated with its previous value.

Conclusion:

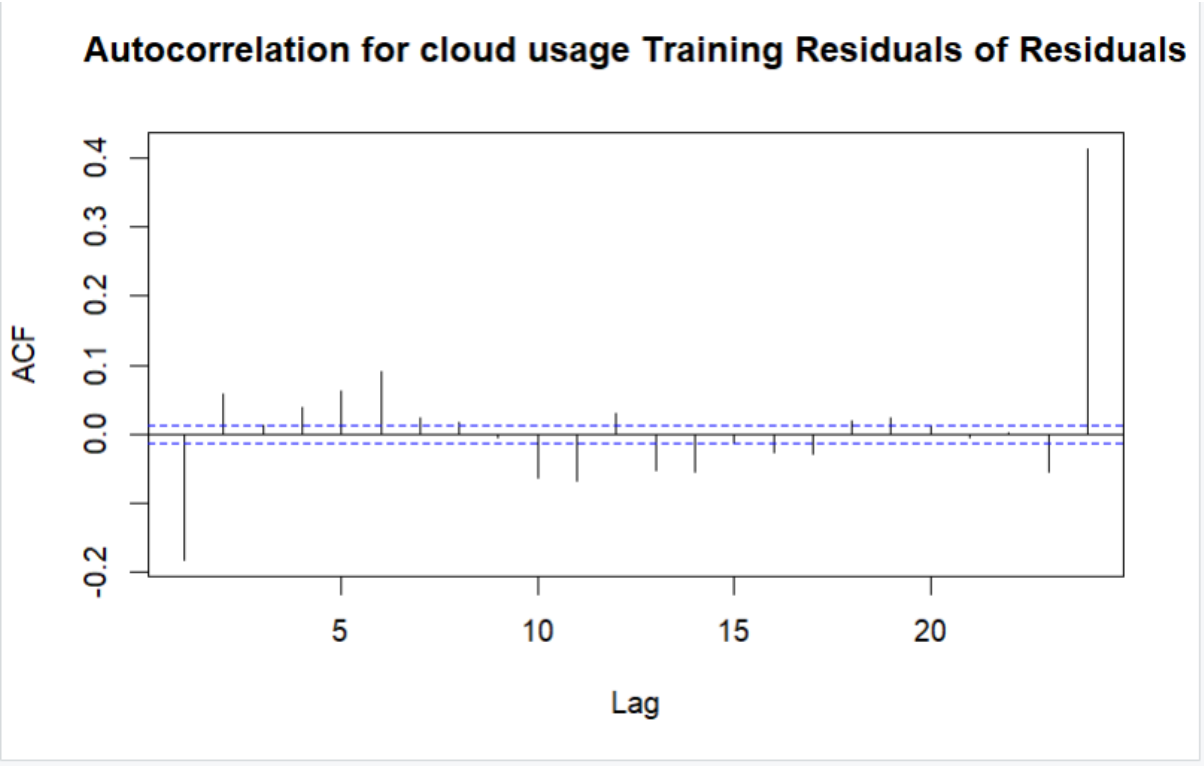
The regression residuals contain significant autocorrelation, which violates the assumption of independence in the linear model. This justifies applying an **AR(1) model to the residuals** as a second-level correction to improve forecast accuracy.

AR(1) Model Fitted to Regression Residuals

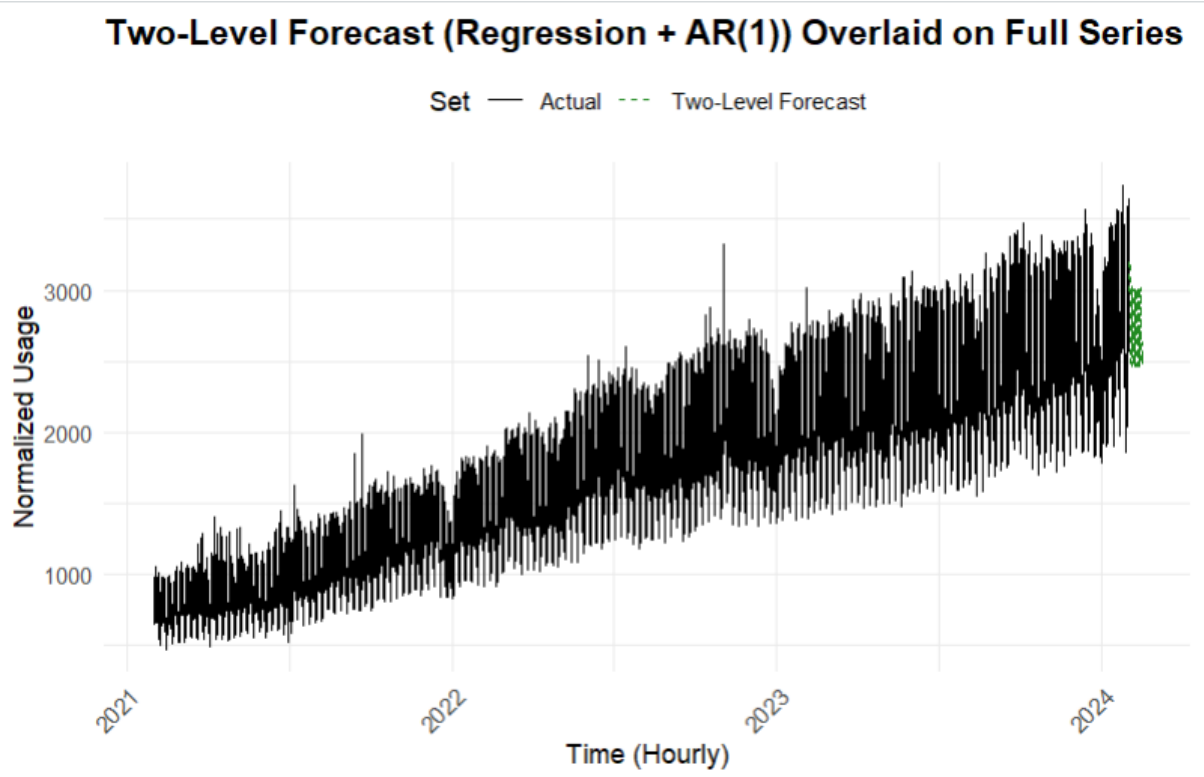
Model Summary and Key Insights

- AR(1) coefficient (ϕ) = 0.9545
  This indicates that approximately 95.45% of the current residual is explained by the immediately preceding residual, showing very strong autocorrelation in the regression residuals.
- Mean = 0.0000
  The residuals have a near-zero mean, confirming that the model does not systematically under- or over-predict.
- Standard error of AR(1) coefficient = 0.0019
  This extremely small value indicates that the AR(1) coefficient is estimated with very high precision and is statistically significant.
- $Sigma^2$ = 6,169
  This is the estimated variance of the residuals after fitting the AR(1) model. It reflects the remaining random noise not captured by the model.
- AIC = 288395.6, AICc = 288395.6, BIC = 288420
  These are information criteria used to compare models. While not interpretable on their own, lower values indicate a better model when comparing alternatives (e.g., ARIMA vs AR(1)).
- ACF1 = -0.1822
  The autocorrelation at lag 1 in the residuals after fitting the AR(1) model is close to zero and slightly negative. This suggests that the AR(1) model has successfully removed the residual autocorrelation, producing white noise-like residuals.

The AR(1) model effectively captures the remaining autocorrelation structure in the regression residuals. It is a strong candidate for residual correction in a two-level forecasting framework and helps enhance the model's short-term predictive power.



The above plot shows the Ar(1) model has captured significant Auto correlation.



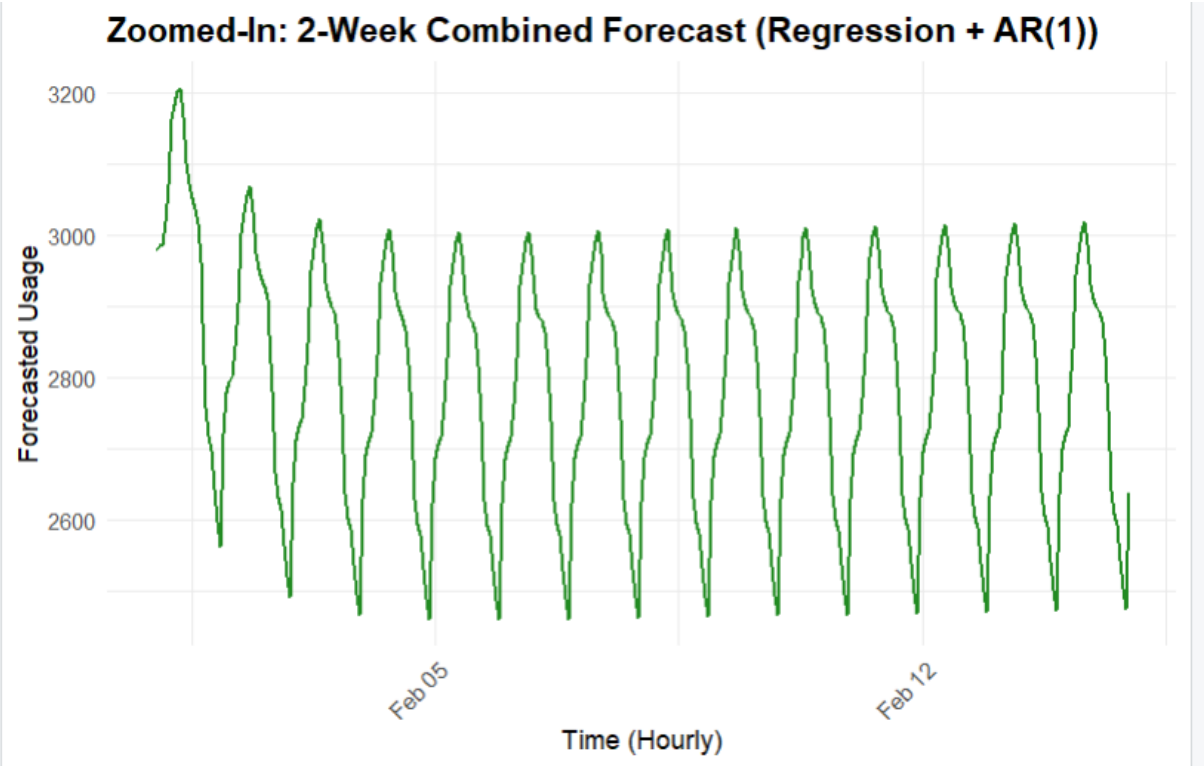


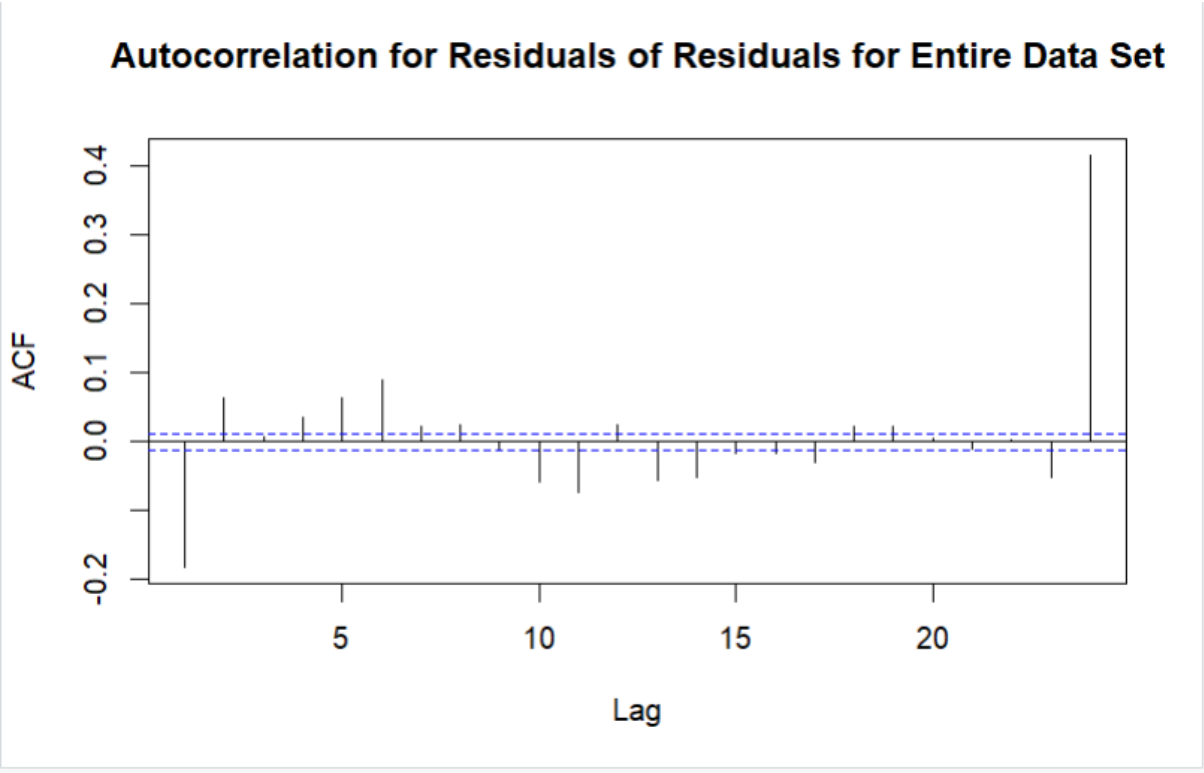
Figure: 2-Week Combined Forecast – Linear Regression + AR(1)

This plot displays the hourly forecast for cloud usage over a 2-week horizon using a two-level forecasting model that combines:

- Linear regression with trend and daily seasonality
- An AR(1) model applied to the regression residuals

Key Observations:

- The forecast captures strong daily cycles, with a consistent pattern of peaks and troughs repeating every 24 hours.
- The initial spike at the beginning of the forecast window reflects momentum captured by the AR(1) correction this adjusts for short-term autocorrelation in the residuals.
- After the first day, the pattern stabilizes, showing smoother daily cycles, which is typical for regression-driven forecasts combined with short-memory corrections.
- The forecasted demand values fluctuate between approximately 2600 and 3200 normalized units, offering fine-grained, hour-level projections ideal for short-term capacity planning.



This plot confirms that the AR(1) model has successfully captured the autocorrelation structure that remained after the initial regression. Therefore, combining the regression model with an AR(1) model for residuals forms a strong two-level forecasting model. This combined approach adequately accounts for trend, seasonality, and residual correlation.

Auto Arima Model:

ARIMA(3,0,2)(2,1,0)[24] Model Overview

This ARIMA model was automatically selected to forecast hourly cloud usage and is expressed in the form:

ARIMA(p,d,q)(P,D,Q)[s]

Where:

- p = 3: Includes 3 non-seasonal autoregressive (AR) terms — the model uses the past 3 values to predict the current value.
- d = 0: No non-seasonal differencing is needed; the original data is already stationary.
- q = 2: Includes 2 non-seasonal moving average (MA) terms — uses past 2 forecast errors to improve accuracy.

- P = 2: Includes 2 seasonal AR terms — accounts for recurring hourly patterns across daily cycles.
- D = 1: Applies one seasonal difference at lag 24 (i.e., subtracts the value from the same hour the day before) to remove seasonal effects.
- Q = 0: No seasonal moving average terms are included.
- s = 24: The seasonal cycle length is 24, corresponding to hourly data with daily seasonality.

Parameter and value	**Interpretation**
ar1 = 2.4401, ar2 = –2.0677, ar3 = 0.6195	**These AR terms capture complex non-seasonal dependencies across the last 3 hours. The pattern suggests some non-linear interaction in hourly demand.**
ma1 = –1.6388, ma2 = 0.8568	**These MA terms adjust forecasts using past residual errors. The large negative MA(1) term strongly corrects for the previous hour's error.**
sar1 = –0.5029, sar2 = –0.2798	**These seasonal AR terms capture the daily cycle by relating current demand to the same hour on previous days. The negative signs indicate an inverse seasonal pattern.**
drift = 0.0845	**Represents a small but steady upward trend in usage across time, even after differencing.**

Metric	**Value**	**Meaning**
RMSE	72.07	Forecast error (in normalized usage units) is relatively low
MAE	53.89	Small absolute error per hour
MAPE	3.36%	Excellent accuracy forecasts are on average within 3.36% of actual values
ACF1	–0.041	Little residual autocorrelation residuals are approximately white noise

ARIMA(3,0,2)(2,1,0)[24] Model Summary- Full dataset

The selected model is ARIMA(3,0,2)(2,1,0)[24], which includes both non-seasonal and seasonal components suitable for hourly data with daily seasonality (period = 24).

Model Structure

- p = 3: Includes 3 non-seasonal autoregressive (AR) lags.
- d = 0: No non-seasonal differencing needed.
- q = 2: Includes 2 non-seasonal moving average (MA) terms.
- P = 2: Includes 2 seasonal autoregressive terms.
- D = 1: Applies one seasonal difference to handle daily seasonality.
- Q = 0: No seasonal MA terms.
- s = 24: Seasonal period of 24 (hourly data with a daily cycle).

Component	Coefficients	Interpretation
AR terms	ar1 = 2.4364, ar2 = –2.0577, ar3 = 0.6133	Reflects non-linear influence from the previous 3 hours.
MA terms	ma1 = –1.6377, ma2 = 0.8553	Past forecast errors are used to adjust predictions.
Seasonal AR terms	sar1 = –0.5013, sar2 = –0.2758	Daily patterns captured through hourly values from 1 and 2 days ago.

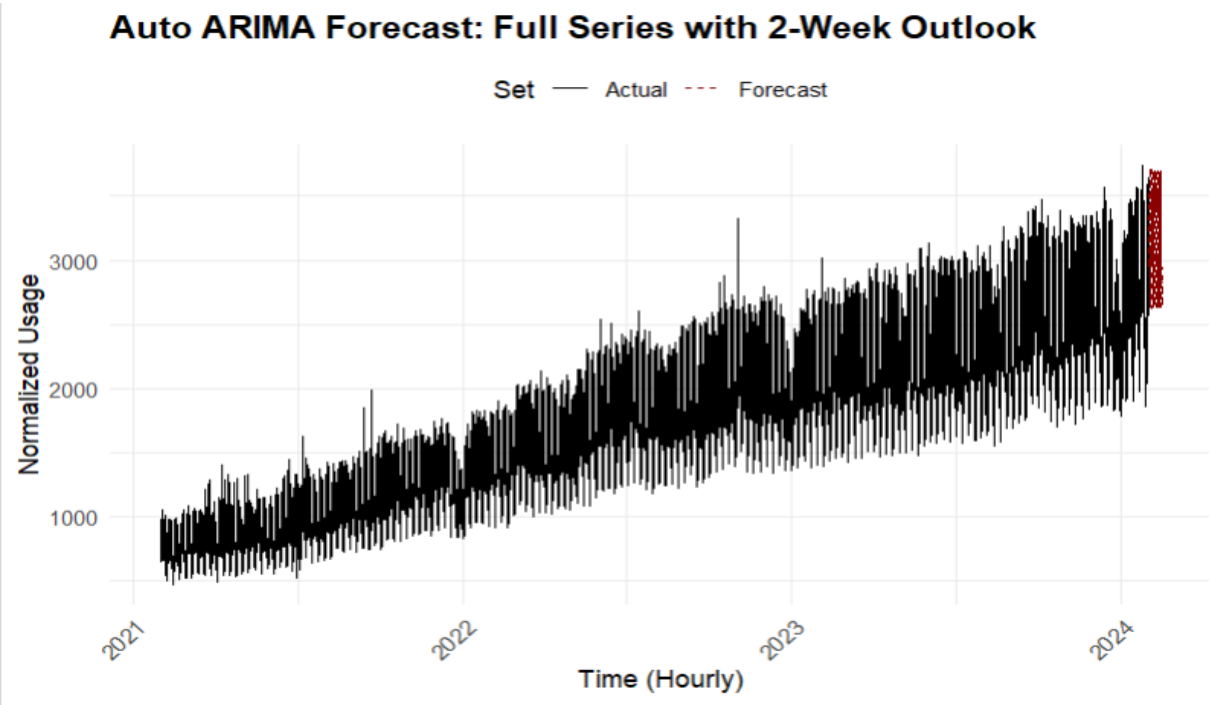


Figure: Auto ARIMA Forecast – Full Series with 2-Week Outlook

This plot displays the **entire hourly cloud usage time series** (black line) along with a **2-week-ahead forecast** (red dashed line) generated by the **Auto ARIMA** model.

Key Observations:

- The **black line** represents historical usage from **February 2021 to January 2024**, showing a strong **upward trend** and high-frequency fluctuations.
- The **red dashed line** at the far right shows the **Auto ARIMA forecast for the next 2 weeks** (336 hours).
- The forecast follows the established trend and aligns well with the recent level of demand, demonstrating the model's ability to **capture both trend and daily seasonality**.
- The usage pattern shows consistent **daily cycles**, which the model incorporates through seasonal autoregressive components.

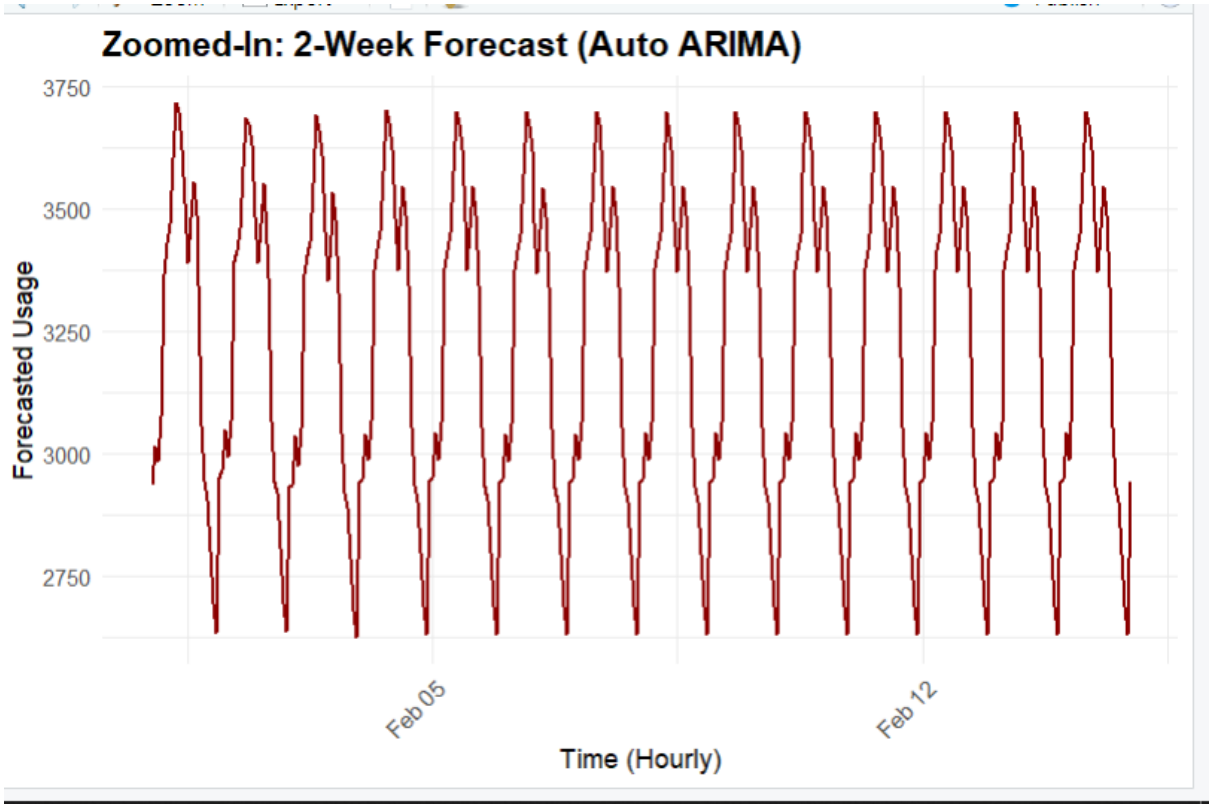


Figure: Zoomed-In 2-Week Forecast (Auto ARIMA)

This plot shows the forecasted hourly cloud usage over a 2-week horizon, generated by the Auto ARIMA(3,0,2)(2,1,0)[24] model.

Key Observations:

- The forecast exhibits a strong and consistent daily pattern, with regular peaks and troughs repeating every 24 hours.
- The usage fluctuates between approximately 2,750 and 3,750 normalized units, capturing high-frequency seasonal behavior.
- The pattern is stable and predictable, indicating that the ARIMA model has effectively captured both the short-term dynamics and daily seasonality through its autoregressive and seasonal components.
- There are no visible anomalies or outliers, and the forecast shows smooth continuity with the observed data leading into the forecast period.

Step 8 Final Model Accuracy Comparison:

Model	RMSE	MAPE (%)	Key Observation
Seasonal Naïve (snaive)	279.42	10.82	Weak performance baseline with high error; unable to capture trend or seasonality.
Holt-Winters (ETS)	66.99	3.04	Strong model with low error; effectively captures level, trend, and daily cycles.
Linear Regression (Trend + Seasonality)	271.35	13.3	High error due to residual autocorrelation; lacks short-term correction.
Two-Level Model (Reg + AR(1))	80.06	3.76	Very good performance; AR(1) improves accuracy by capturing residual patterns.
Auto ARIMA	73.18	3.33	Best balance of performance and structure; captures both short-term and seasonal patterns.

Conclusion:

- Auto ARIMA and Holt-Winters performed best overall with lowest MAPE and RMSE, making them strong candidates for operational forecasting.
- The two-level model provided solid results with interpretability benefits and decent accuracy.

- Seasonal Naïve serves as a useful baseline but performs poorly compared to statistical models.

Top 3 Forecasting Models: Accuracy Comparison

1. Holt-Winters (ETS) – Rank 1

- **RMSE**: 66.99
- **MAPE**: 3.04%
- **ACF1**: –0.007

Strengths:

- Achieved the **lowest RMSE and MAPE**, indicating highly accurate forecasts.
- Residuals show near-zero autocorrelation, meaning model structure is well-fitted.
- Captures level and trend effectively with minimal complexity.
- Suitable for short-term operational forecasts.

Limitations:

- Does not explicitly model autocorrelation in residuals.
- Assumes **fixed** seasonality, which may not adapt well if daily patterns shift over time.

2. Auto ARIMA – Rank 2

- **RMSE**: 73.18
- **MAPE**: 3.33%
- **ACF1**: –0.041

Strengths:

- Captures both **short-term autocorrelation** and **daily seasonality** through non-seasonal and seasonal terms.
- Automatically selects optimal model structure using AIC/BIC.
- Residuals are close to white noise, indicating good fit.

Limitations:

- Slightly higher error than ETS.
- More complex and harder to interpret due to multiple AR and MA components.

- Can be sensitive to outliers or structural changes.

3. Two-Level Model (Regression + AR(1)) – Rank 3

- **RMSE**: 80.06
- **MAPE**: 3.76%
- **ACF1**: –0.182

Strengths:

- Combines the interpretability of regression with short-term dynamics via AR(1).
- Reduces autocorrelation present in regression-only residuals.
- Useful for explainable forecasting — ideal in business or policy reporting.

Limitations:

- Accuracy is lower than ETS and Auto ARIMA.
- Slightly more complex pipeline due to multi-step model fitting.
- May struggle if seasonality is irregular or if residuals are non-linear.

Business Implications (Based on Holt-Winters Forecast)

Based on our forecast using the **Holt-Winters exponential smoothing model**, cloud compute demand is expected to increase by approximately **15.86%** over the next two weeks, rising from an average of **3,020** to **3,497 normalized hourly usage units**.

This forecast carries several key business implications across infrastructure, operations, and strategic planning domains:

Capacity Planning & Procurement

Cloud infrastructure teams can use this short-term forecast to **pre-provision compute resources** such as virtual machines, containers, or serverless quotas, ensuring adequate capacity during projected demand surges. This helps prevent **service slowdowns** or **resource contention** under load.

Cloud Cost Optimization

With usage projected to rise steadily, finance and procurement teams can **optimize cloud spend** by securing **reserved instances** or leveraging **spot market purchases** in advance. This reduces cost volatility and improves cost efficiency per compute unit.

Autoscaling and SLA Management

Reliable hourly-level forecasts allow platform and SRE teams to fine-tune **autoscaling policies**. Forecast-driven scaling minimizes cold starts and helps maintain **application performance and SLA compliance**, especially during periods of peak demand.

Energy and Sustainability Impact

Increased compute demand directly affects **power consumption and cooling requirements**. Forecasts enable data center managers to **anticipate energy loads**, plan **renewable energy allocation**, and stay aligned with **sustainability and carbon reduction goals**.

Strategic IT Budgeting

Forecasted demand growth informs **cloud budgeting** and **resource planning** across departments. By aligning IT budgets with usage trends, organizations can **prevent overspending** while maintaining operational readiness.

Conclusion

The 15.86% projected rise in hourly compute demand emphasizes the importance of accurate, data-driven forecasting models. With Holt-Winters delivering high accuracy (MAPE ≈ 3.0%), this model supports **tactical planning** and **resource optimization** in large-scale, dynamic cloud environments.

References and Resources

Dataset

- Snowflake Labs – Shaved Ice Normalized Cloud Usage Dataset
  https://github.com/Snowflake-Labs/shavedice-dataset
  Open-source dataset used in this project. Contains normalized hourly cloud compute usage across regions and VM types.

R Forecasting Packages & Documentation

- forecast package (CRAN)
  https://cran.r-project.org/web/packages/forecast/forecast.pdf
  Official documentation for functions like tslm(), ets(), auto.arima(), forecast(), and Acf().
- Forecasting: Principles and Practice (Hyndman & Athanasopoulos)
  https://otexts.com/fpp3/
  A foundational textbook on time series forecasting using R, freely available online.
- R documentation for tslm()
  https://pkg.robjhyndman.com/forecast/reference/tslm.html

STL Decomposition

- STL in Statsmodels (Python docs, conceptually aligned with R)
  https://www.statsmodels.org/dev/generated/statsmodels.tsa.seasonal.STL.html
  STL (Seasonal-Trend decomposition using Loess) description and implementation.

Business Applications of Forecasting

- Google Cloud: Capacity Planning with Forecasting
  https://cloud.google.com/blog/products/ai-machine-learning/forecasting-inventory-demand-with-time-series-data
  Real-world example of using time series for cloud infrastructure planning.
- AWS EC2 Auto Scaling Documentation
  https://docs.aws.amazon.com/autoscaling/ec2/userguide/what-is-amazon-ec2-auto-scaling.html
  Describes how infrastructure responds to demand using forecast-driven autoscaling.

Next Steps and Future Work

Building on the insights and performance of the current forecasting models, several extensions and refinements are planned for future exploration:

Implementation of Advanced Models

- Prophet Model:
  Incorporate Facebook's Prophet model, known for its flexibility in handling seasonality, holidays, and change points. It will be tested alongside ETS and ARIMA to assess its adaptability to cloud demand cycles.

- Ensemble Forecasting:
  Explore ensemble methods that combine forecasts from multiple models (e.g., ETS, ARIMA, Prophet, and regression) to improve overall robustness and accuracy.

Granular and Segment-Level Forecasting

- Region-Specific Modeling:
  Rather than modeling aggregate demand, future work will involve building individual models for each cloud region or VM instance type to capture localized usage patterns and regional seasonality.
- Temporal Segmentation Analysis:
  Perform comparative analysis of cloud usage trends during pre-COVID, COVID, and post-COVID periods to assess how macroeconomic events influenced infrastructure demand patterns.

These enhancements will not only deepen the understanding of demand behavior but also make the forecasting framework more actionable, localized, and adaptable for real-world cloud infrastructure management.